

0

# TK-80 応用プログラム

0

**NEC** 日本電気株式会社

## 目 次

第1章 デジタル・タイマ .....	1
1.1 概 要 .....	1
1.2 概略のフローチャート .....	1
1.3 詳細なフローチャート .....	2
1.4 コーディング例 .....	4
1.5 プログラミングおよび実行方法 .....	5
1.6 タイマ・サブルーチンの時定数の調整 .....	6
第2章 電子サイレン .....	9
2.1 概 要 .....	9
2.2 概略のフローチャート .....	9
2.3 詳細なフローチャート .....	10
2.4 コーディング例 .....	12
2.5 オーディオ・アンプの接続方法 .....	13
2.6 プログラミングおよび実行方法 .....	13
2.7 周波数帯域の変更 .....	14
第3章 プログラマブル・メトロノーム .....	15
3.1 概 要 .....	15
3.2 概略のフローチャート .....	15
3.3 詳細なフローチャート .....	15
3.4 コーディング例 .....	18
3.5 プログラミングおよび実行方法 .....	19
第4章 電子オルガン .....	21
4.1 概 要 .....	21
4.2 フローチャート .....	21
4.3 コーディング例 .....	26
4.4 プログラミングおよび実行方法 .....	28
4.5 キーボードと音階との対応 .....	29
第5章 音楽の自動演奏プログラム .....	31
5.1 概 要 .....	31
5.2 概略のフローチャート .....	31
5.3 詳細なフローチャート .....	32
5.4 コーディング例 .....	34
5.5 楽譜データの作成 .....	35

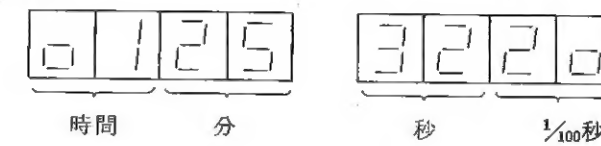
第6章 無限音階プログラム .....	39
6.1 概 要 .....	39
6.2 概略のフローチャート .....	40
6.3 詳細なフローチャート .....	42
6.4 コーディング例 .....	50
6.5 オーディオ・アンプの接続方法 .....	52
6.6 プログラミングおよび実行方法 .....	52

## 第1章 デジタル・タイマ

### 1.1 概 要

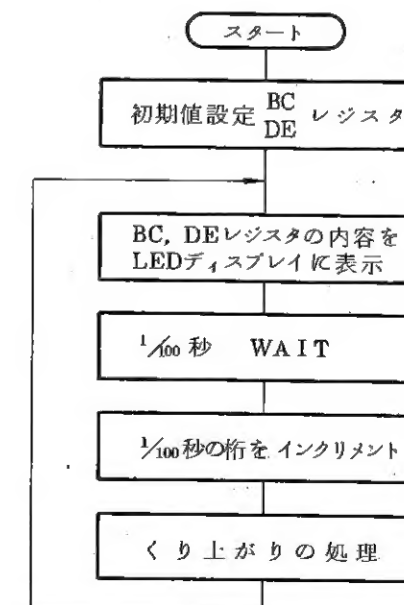
TK-80 のLEDディスプレイを利用して、デジタル・タイマを構成するプログラム例を示します。

LEDディスプレイに表示されるデータは次のようになります。



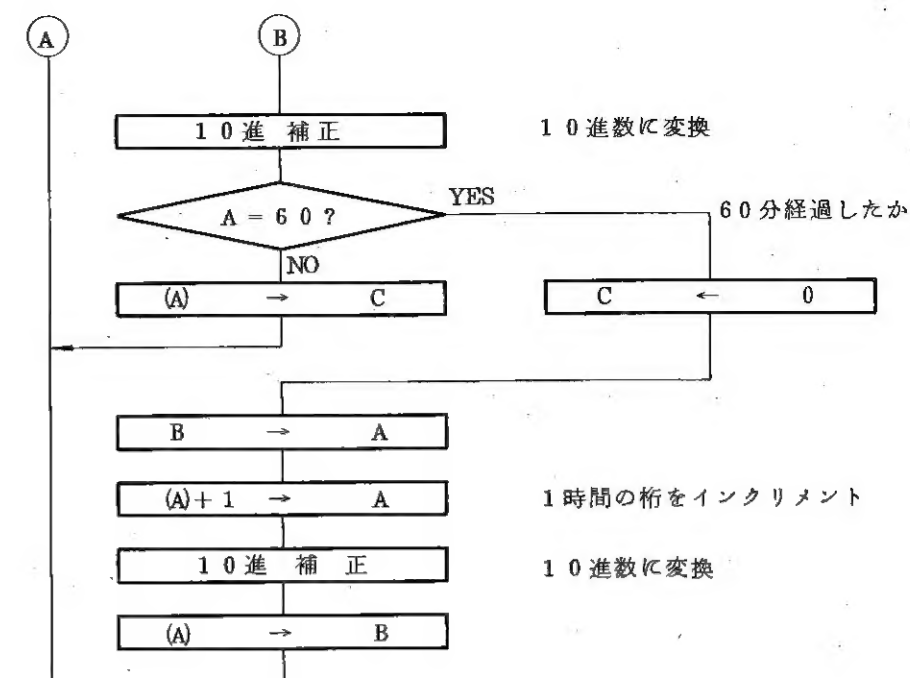
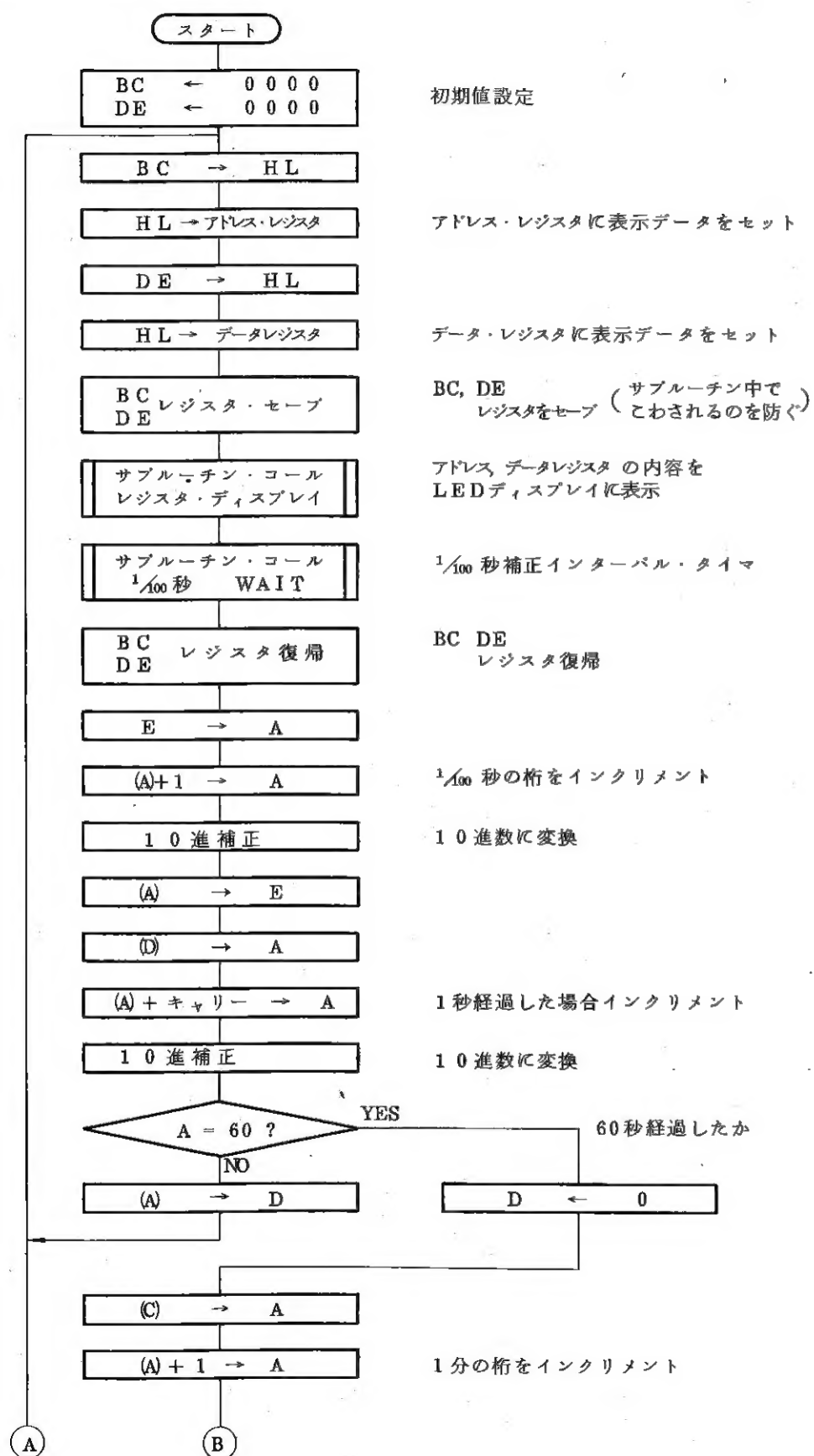
このプログラムは、各単位からの繰り上り処理するメイン・ルーチンと最小単位である  $1/100$  秒をカウントするタイマ・サブルーチンから構成されています。LEDディスプレイへの表示は、モニタ・プログラム内のサブルーチンにより行われます。(モニタ・プログラムのサブルーチンについてはTK-80 ユーザース・マニュアルに記載されています。)

### 1.2 概略のフローチャート

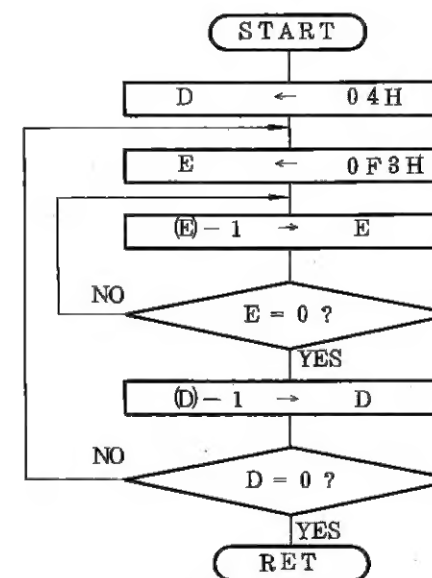


### 1.3 詳細なフローチャート

(1) メイン・ルーチン



(2)  $\frac{1}{100}$  秒インターバルのための補正ルーチン



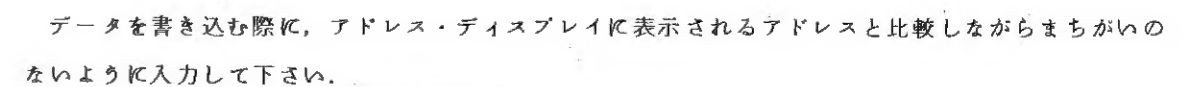
このサブルーチンは、 $\frac{1}{100}$ 秒をカウントアップする際のインターバルをほぼ $\frac{1}{100}$ 秒に補正するためのものでD・Eレジスタにセットされる値により、このタイマーの時定数を変えることができます。

## 4

5

## 1.5 プログラミングおよび実行方法

1.4 に書かれているオブジェクトコードを所定の番地のメモリに書き込んで行きます。



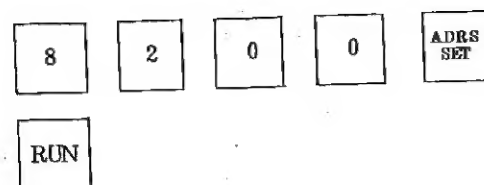
プログラムが終了したら 

READ INCR
--------------

READ DECR
--------------

 キーによりメモリの内容を読み出し、まちがいがあった場合はそのつど訂正して下さい。

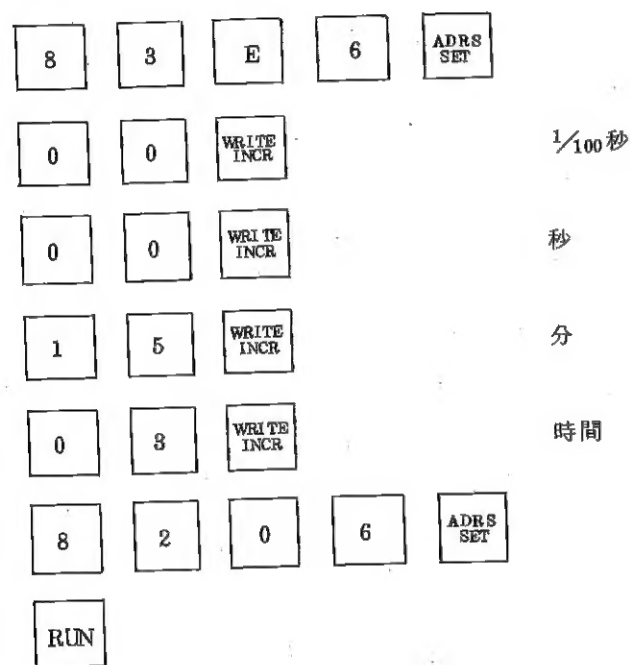
プログラムにまちがいがないことを確認したら次の要領で実行させます。



この操作によりデジタル・タイマーがスタートします。

8200番地からこのプログラムをスタートすると必ず初期値は0となり、0時、0分、0秒からカウントが始まります。

この初期値をあらかじめ設定したい場合は次のように操作して下さい。



上記の例では

0 3 時 1 5 分 0 0 秒 0 0 1/100 秒 よりカウントが始まります。

## 1.6 タイマ・サブルーチンの時定数の調整

本プログラムの基本となっている1/100 秒の時間間隔は、プログラム中のWAITサブルーチンで作られています。

1/100 秒のインターバルは、正確には、コーディング例のラインナンバー10 のサブルーチンコールによってデータが表示されてからメイン・ルーチンを一周し、次のデータが同様に示して表示されるまでの時間ということになりますが、これらの時間のうち大部分がWAITサブルーチン中で費やされています。これらの時間は、各命令をCPUが実行する際に費やされる時間とDMA転送によってCPUの処理が中断される時間との合計ということになります。よってDMA転送による誤差(極度

1 msec に10μ sec程度)を考慮してタイマサブルーチンのパラメータを変更すればいろいろなインターバルタイムを作り出すことができます。

実際にマイクロプロセッサによってタイマを作る場合は、上記のようなDMA転送は行わないためメインクロックの精度に相当する精度でタイマを作ることができます。CPUが1つの命令を実行する時に費やされる時間はμCOM-80 ユーザーズ・マニュアル(IEM-533B)の86ページから98ページに掲載されているタイミング・ダイアグラムより計算することができます。

以下にその1例を示します。

MVI 命令

(MVI D, 04H 等)

μCOM-80 ユーザーズ・マニュアル

TK-80 で実行した場合

M1(マシンサイクル1)				M2(マシンサイクル2)				M1				M2			
T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	T <sub>4</sub>	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	T <sub>4</sub>	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	T <sub>4</sub>	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	T <sub>4</sub>

MVI 命令は2つのマシンサイクルで構成され、M1は4つのステート、M2は3つのステートで構成されています。

TK-80 でこの命令を実行した場合、メモリからのデータ待合わせのため、T<sub>2</sub> のみが2回くり返され右図のようにM1は5つ、M2は4つのステートで構成されることになります。

又、TK-80 のクロック周波数は約20.5MHzですから、1つのステートを実行するためには約488 nsec かかることになります。

$$488 \times 9 = 4392 \text{ (nsec)}$$

よってMVI 命令を実行するためには約4.4 μsec の時間がかかることになります。

TK-80 で各命令を実行する場合、すべての命令の各マシンサイクル中のT<sub>2</sub> ステートは2回くり返されることになります。

(2番目のT<sub>2</sub> はメモリー・アクセスの待ち時間として動作するようになっています。TK-80 ユーザーズ・マニュアルの第5章(TK-80 のハードウェア)を参考にして下さい。)

## 第2章 電子サイレン

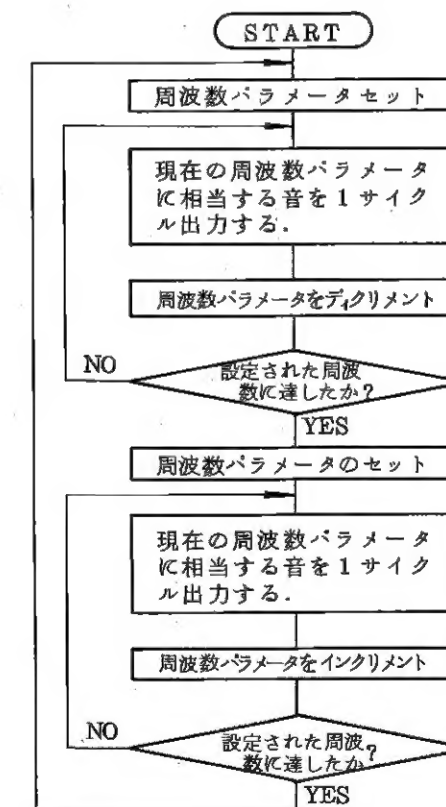
### 2.1 概 要

PPI ( $\mu$ PD 8255)のポートCに接続されたオーディオ・アンプにサイレン音を出力するプログラム例を示します。

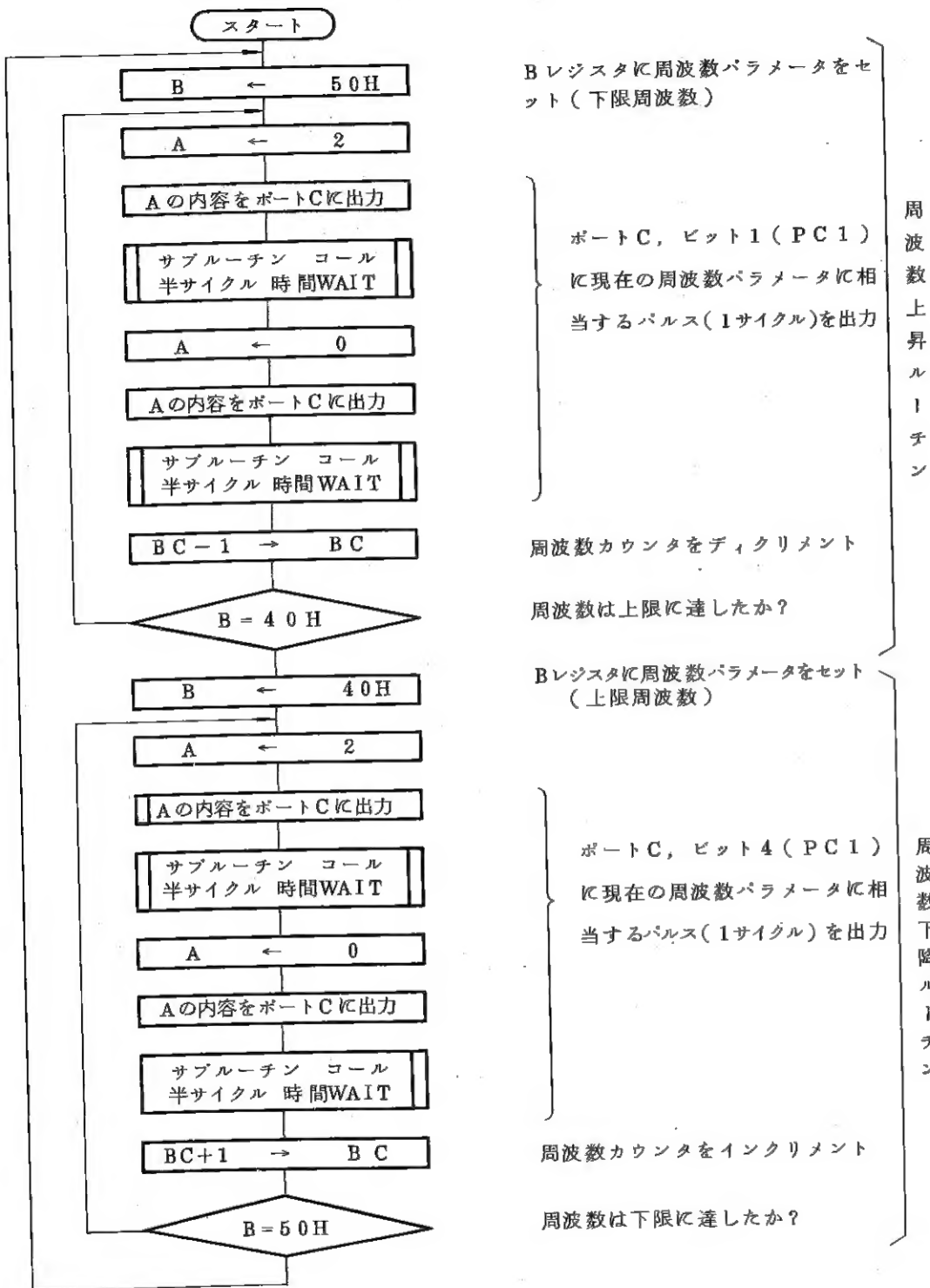
このプログラムは、モニタにより出力ポートにセットされているポートCのビット1 (PC1 15番ピン)に“1”と“0”を交互に出力し、オーディオ帯域の周波数のパルスを作り、これをオーディオ・アンプに入力し音声を発生します。

ここで、発生する音声の周波数はCPUにより制御され、ある周波数範囲を連続的に変化させるようにしてサイレン音を作り出しています。

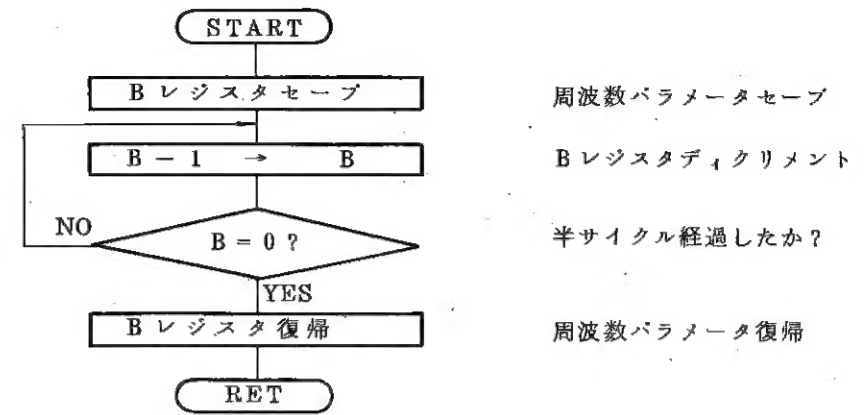
### 2.2 概略のフローチャート



### 2.3 詳細なフローチャート



半サイクル WAIT サブルーチン



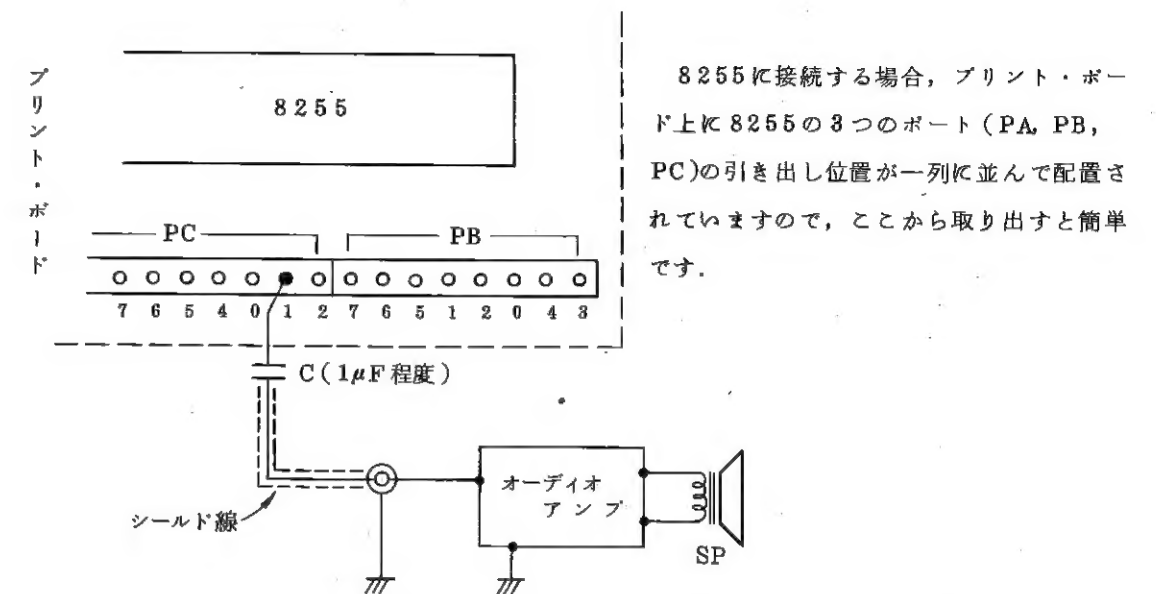


## 2.4 コーディング例

ライン	アドレス	ラベル	ニーモニック	オブジェクトコード
00	8200	START:	MVI B, 50H	06 50
01	02	LOOP1:	MVI A, 2	3E 02
02	04		OUT 2	D3 02
03	06		CALL WAIT	CD 33 82
04	09		MVI A, 0	3E 00
05	0B		OUT 2	D3 02
06	0D		CALL WAIT	CD 33 82
07	10		DCX B	0B
08	11		MOV A, B	78
09	12		CPI 40H	FE 40
10	14		MOV B, A	47
11	15		JNZ LOOP1	C2 02 82
12	18		MVI B, 40H	06 40
18	1A	LOOP2:	MVI A, 2	3E 02
14	1C		OUT 2	D3 02
15	1E		CALL WAIT	CD 33 82
16	21		MVI A, 0	3E 00
17	23		OUT 2	D3 02
18	25		CALL WAIT	CD 33 82
19	28		INX B	03
20	29		MOV A, B	78
21	2A		CPI 50H	FE 50
22	2C		MOV B, A	47
23	2D		JNZ LOOP2	C2 1A 82
24	30		JMP START	C3 00 82
25	33	WAIT:	PUSH B	C5
26	34		DCR B	05
27	35		JNZ \$-1	C2 34 82
28	38		POP B	C1
29	39		RET	C9

## 2.5 オーディオ・アンプの接続方法

オーディオ・アンプはカップリング・コンデンサを介して、PPIのポートC, ピット1 ( $\mu$ PD 8255 15番ピン)に接続します。この端子の開放出力電圧は5V<sub>p-p</sub>です。

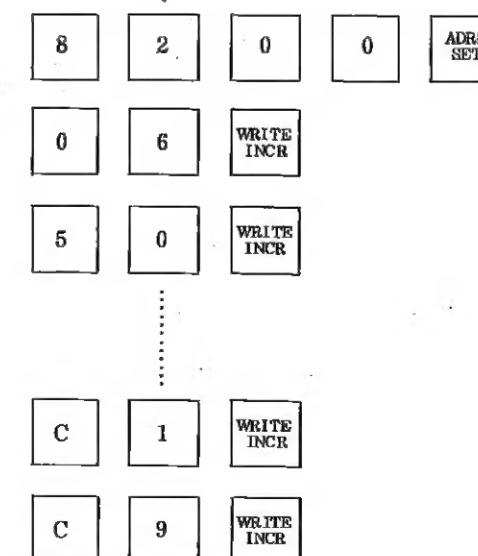


TK-80より引き出すGNDの位置はノイズの最も少ない場所を選んで下さい。

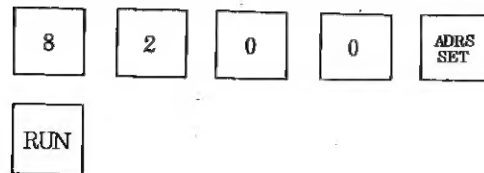
TRラジオのオーディオ・アンプの初段へ入れてやれば充分楽しめます。専用オーディオ・アンプの場合は、LINE入力端子に入力して下さい。

## 2.6 プログラミングおよび実行方法

コーディング・リスト上のオブジェクト・コードを所定のメモリに書き込みます。



プログラムの書込みが終了し、書き込みエラーがないことを確認したら次のコマンドで実行します。



## 2.7 周波数帯域の変更

本プログラムは周波数パラメータ及び上、下限比較データを変更することにより、周波数変化帯域を変更することができます。

周波数パラメータはラインナンバー00及び12において、Bレジスタにセットする値でこの値が大きいほど出力周波数は低くなります。

又、周波数上、下限比較データはラインナンバー09及び21において、アキュムレータと比較しているデータです。

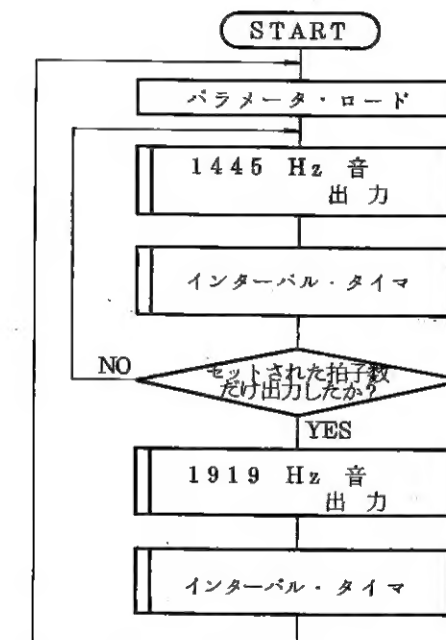
## 第3章 プログラマブル・メトロノーム

### 3.1 概要

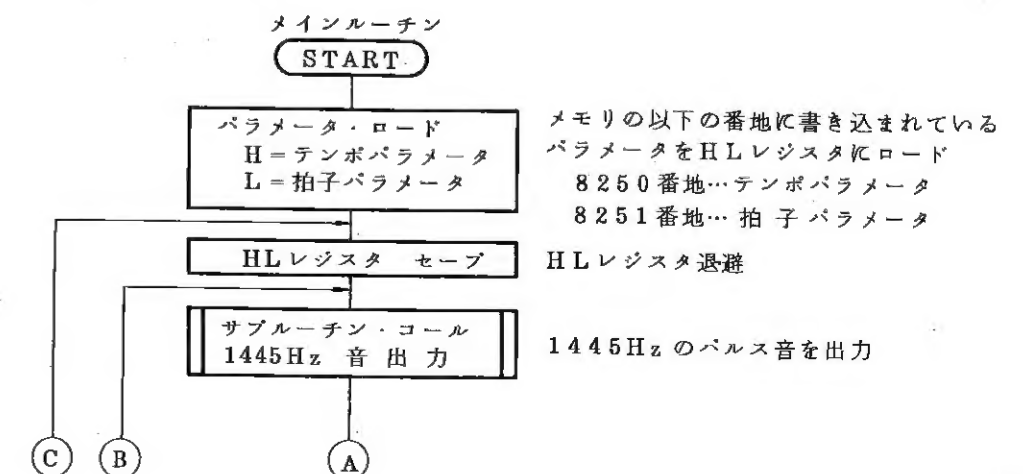
ピアノ、ギター等の練習に使えるプログラマブル・メトロノームのプログラム例を示します。

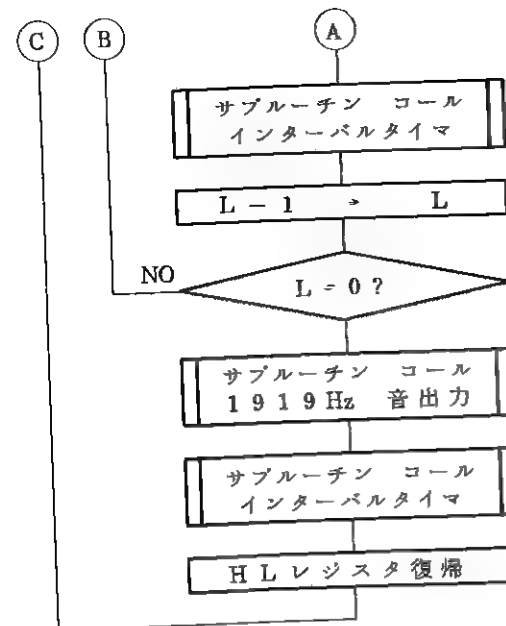
このプログラムはパラメータを2ワードセットすることにより、テンポ、拍子を自由に設定できるメトロノームで、PPI (μPD8255) のポートC、ビット1 (PC1) に可聴周波数のパルスを送り出します。この端子にオーディオアンプを接続すれば音声として聞こえます。

### 3.2 概略のフローチャート



### 3.3 詳細なフローチャート





テンポパラメータによって指定された時間待合わせ

拍子パラメータをディクリメント

1919Hz のパルス音を出力

テンポパラメータによって指定された時間待合わせ

HLレジスタ復帰

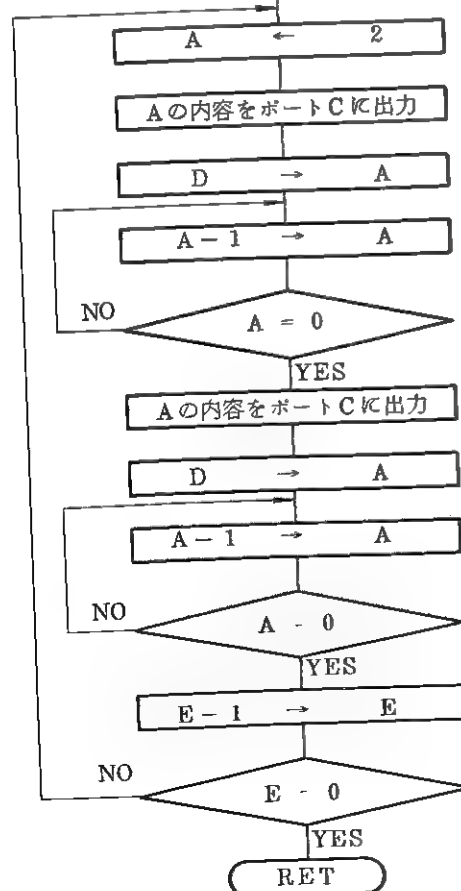
サブルーチン  
1445Hz (T1)  
スタート

レジスタ イニシャライズ  
D = 35H  
E = 12H

音声出力ルーチン  
1919Hz (T2)  
スタート

レジスタ イニシャライズ  
D = 26H  
E = 16H

周波数パラメータ  
セット



ポートC PC1に“1”を出力

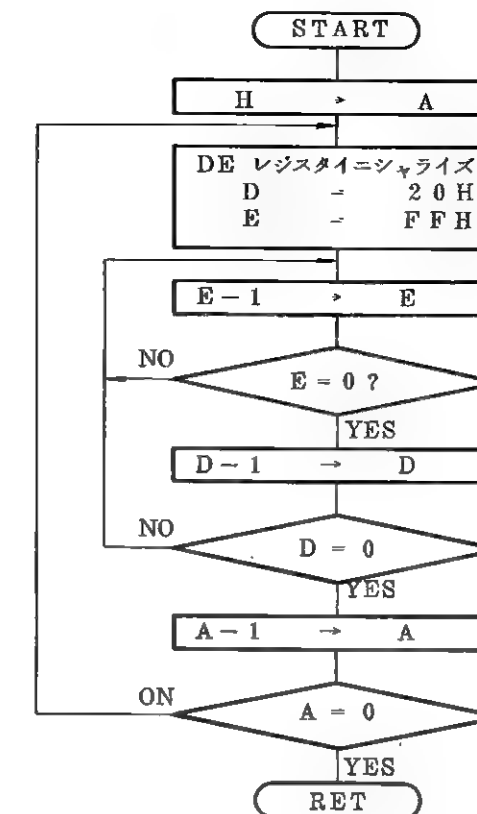
半サイクル時間待合わせ

ポートC PC1に“0”を出力

半サイクル時間待合わせ

各パルス音の長さだけくり返す

# サブルーチン インターバルタイマ (WAIT)



テンポパラメータを アキュムレータにロード

単位時間タイマ用パラメータをDEレジスタにセット

単位時間WAIT ループ

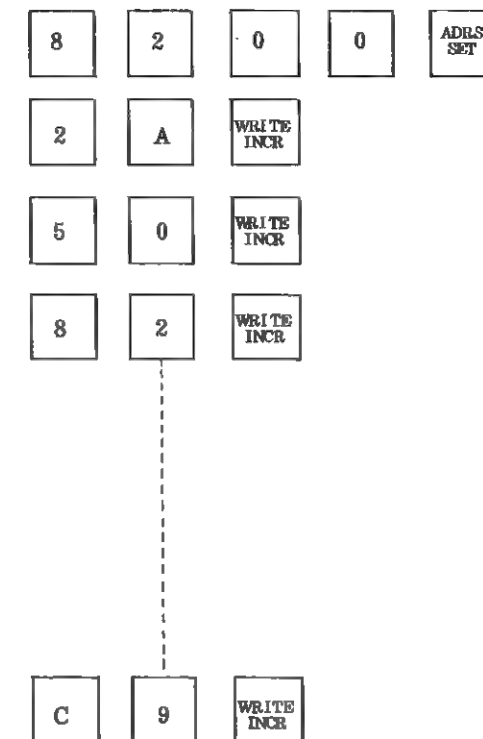
テンポパラメータにセットされた回数だけくり返す。

### 3.4 コーディング例

ライン	アドレス	ラベル	ニーモニック	オブジェクトコード
00	8200	START:	LHLD DATA	2A 50 82
01	08		DCR L	2D
02	04	LOOP1:	PUSH H	E5
03	05	LOOP2:	CALL T1	CD 19 82
04	08		CALL WAIT	CD 87 82
05	0B		DCR L	2D
06	0C		JNZ LOOP2	C2 05 82
07	0F		CALL T2	CD 1F 82
08	12		CALL WAIT	CD 87 82
09	15		POP H	E1
10	16		JMP LOOP1	C3 04 82
11	19	T1 :	LXI D, 8512H	11 12 82
12	1C		JMP T3	C3 22 82
13	1F	T2 :	LXI D, 2616H	11 16 26
14	22	T3 :	MVI A, 2	3E 02
15	24		OUT 2	D3 02
16	26		MOV A, D	7A
17	27		DCR A	3D
18	28		JNZ \$-1	C2 27 82
19	2B		OUT 2	D3 02
20	2D		MOV A, D	7A
21	2E		DCR A	3D
22	2F		JNZ \$-1	C2 2E 82
23	32		DCR E	1D
24	33		JNZ T3	C2 22 82
25	36		RET	C9
26	37	WAIT :	MOV A, H	7C
27	38		LXI D, 20FFH	11 FF 02
28	3B		DCR E	1D
29	3C		JNZ \$-1	C2 3B 82
30	3F		DCR D	15
31	40		JNZ \$-5	C2 3B 82
32	43		DCR A	3D
33	44		JNZ WAIT+1	C2 38 82
34	47		RET	C9

### 3.5 プログラミングおよび実行方法

3.4に書かれているオブジェクトコードを所定の番地のメモリに書き込んで行きます。

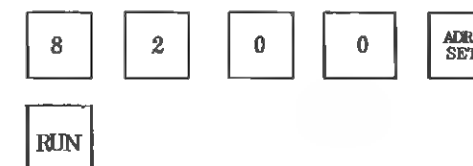


プログラムの書込みが終了し書込エラーがないことを確認したならば、テンポと拍子パラメータをセットします。

テンポパラメータは8250番地で00~FFまでのデータを設定できます。データは大きな値にするほどテンポはおそくなります。

拍子パラメータは8251番地で02~FFまでのデータを設定できます。2拍子にしたい場合は'02' 3拍子にしたい場合は'03'というぐあいにセットします。

パラメータのセットが終了すると次のキーコマンドで実行します。



なお、オーディオアンプの接続法は2.5と全く同じです。

## 第4章 電子オルガン

### 4.1 概 要

TK-80 のキーボードをオルガンの鍵盤として使用した電子オルガンのプログラム例を示します。  
このプログラムは、音階をソフトウェアで作出し、PPI ( $\mu$ PD8255)のポートCのPC<sub>1</sub>に音声帯域のパルスとして出力するもので、カプリング・キャパシタを介してオーディオアンプに接続することにより、電子オルガンを構成することができます。

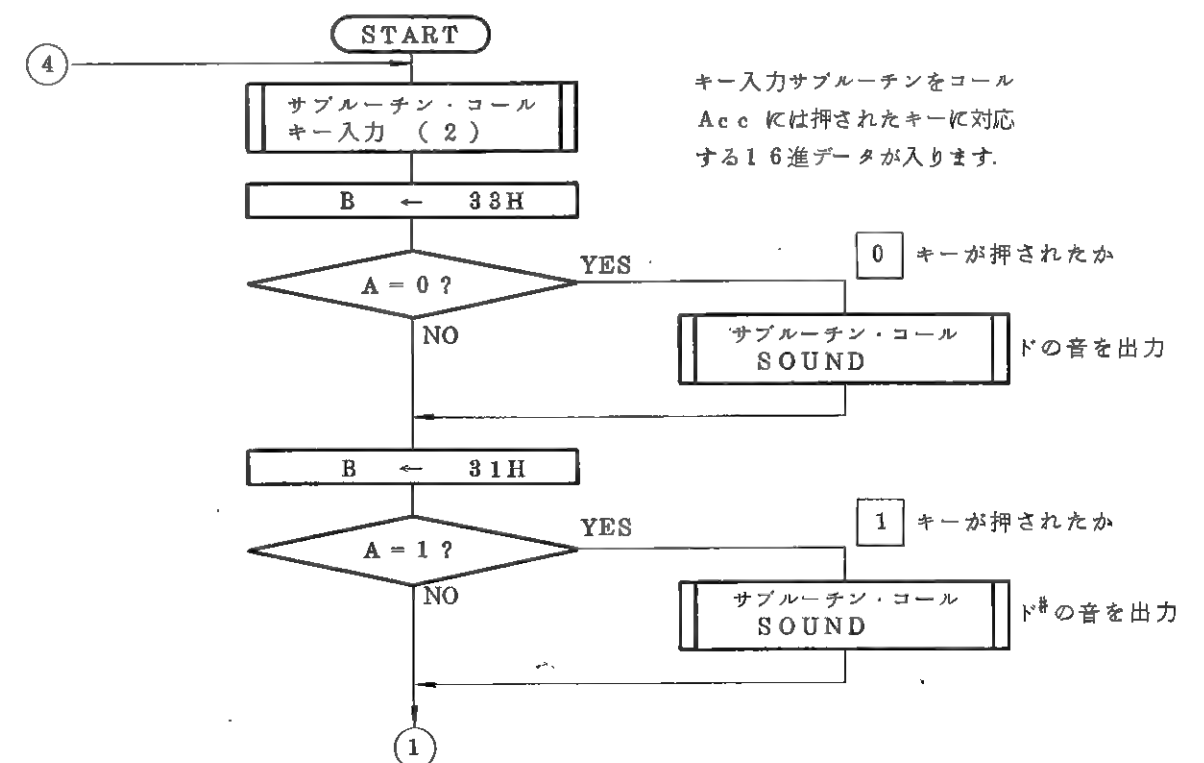
このプログラムはキーセンス用のメインルーチンと、実際に各音階に相当する音を発生させるサブルーチン (SOUND) によって構成されています。

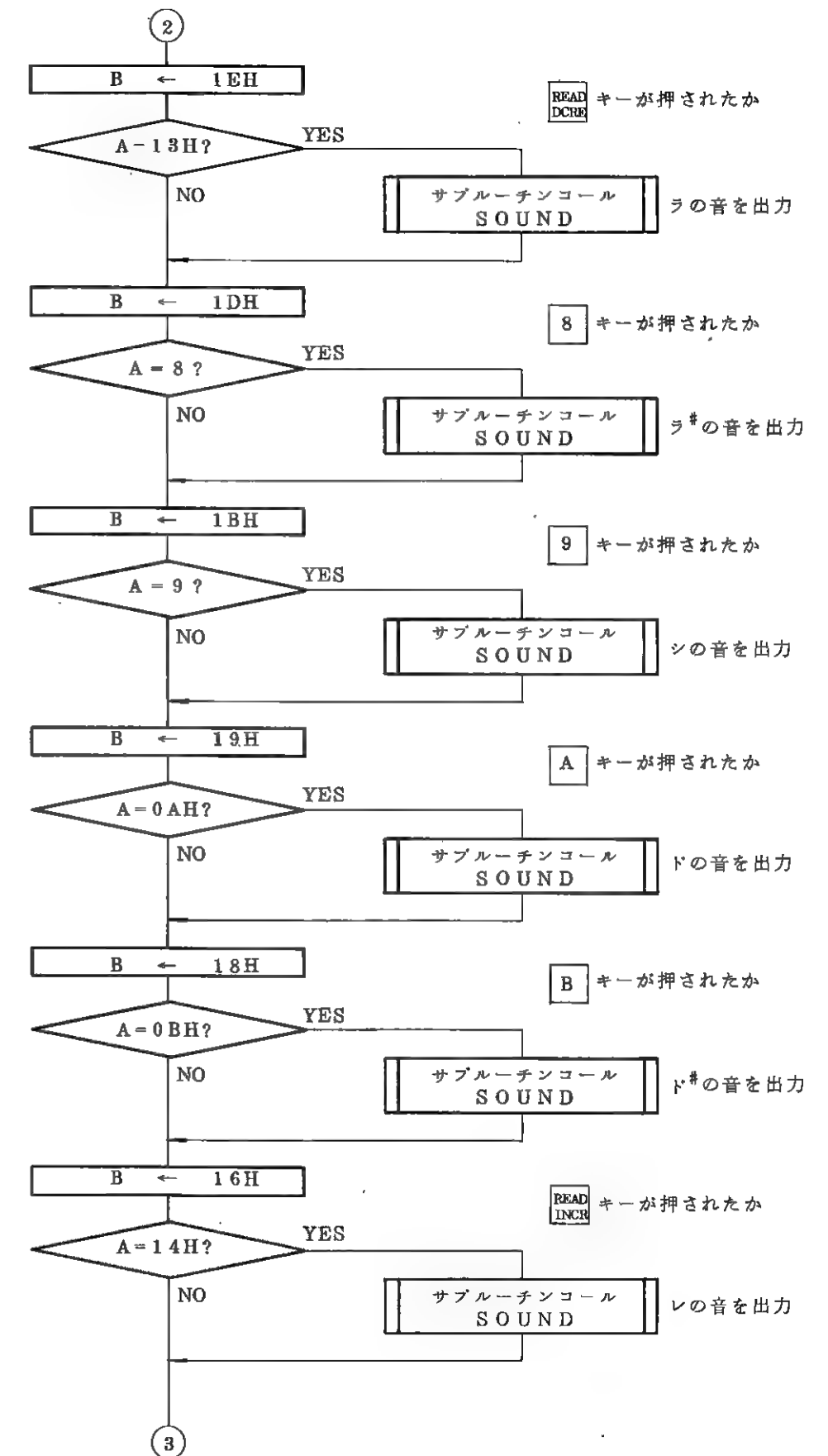
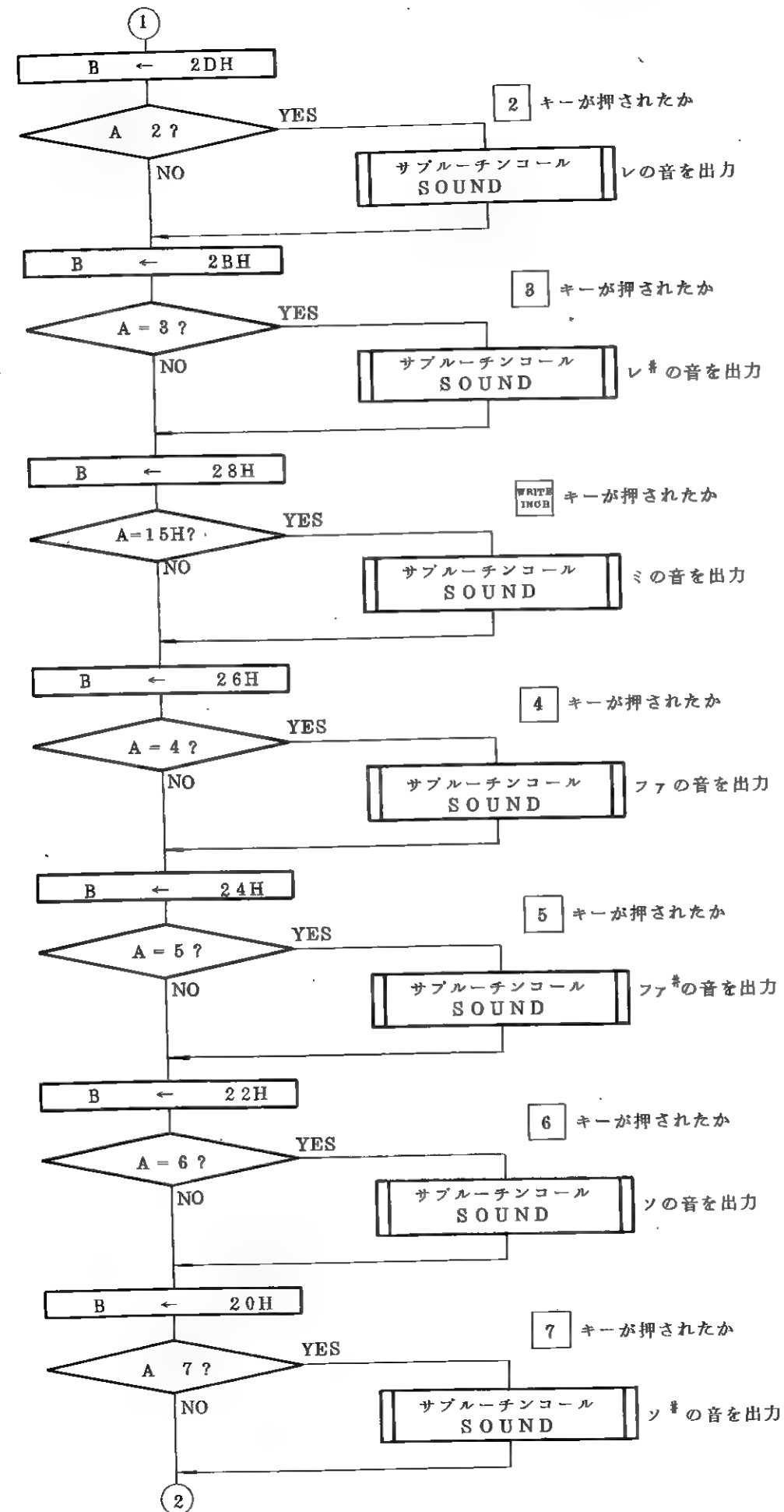
メインルーチンでBレジスタに各音階に相当する周波数パラメータをセットして、サブルーチン SOUNDをコールすることによりそのパラメータの周波数に相当する音を一定時間発生させることができます。

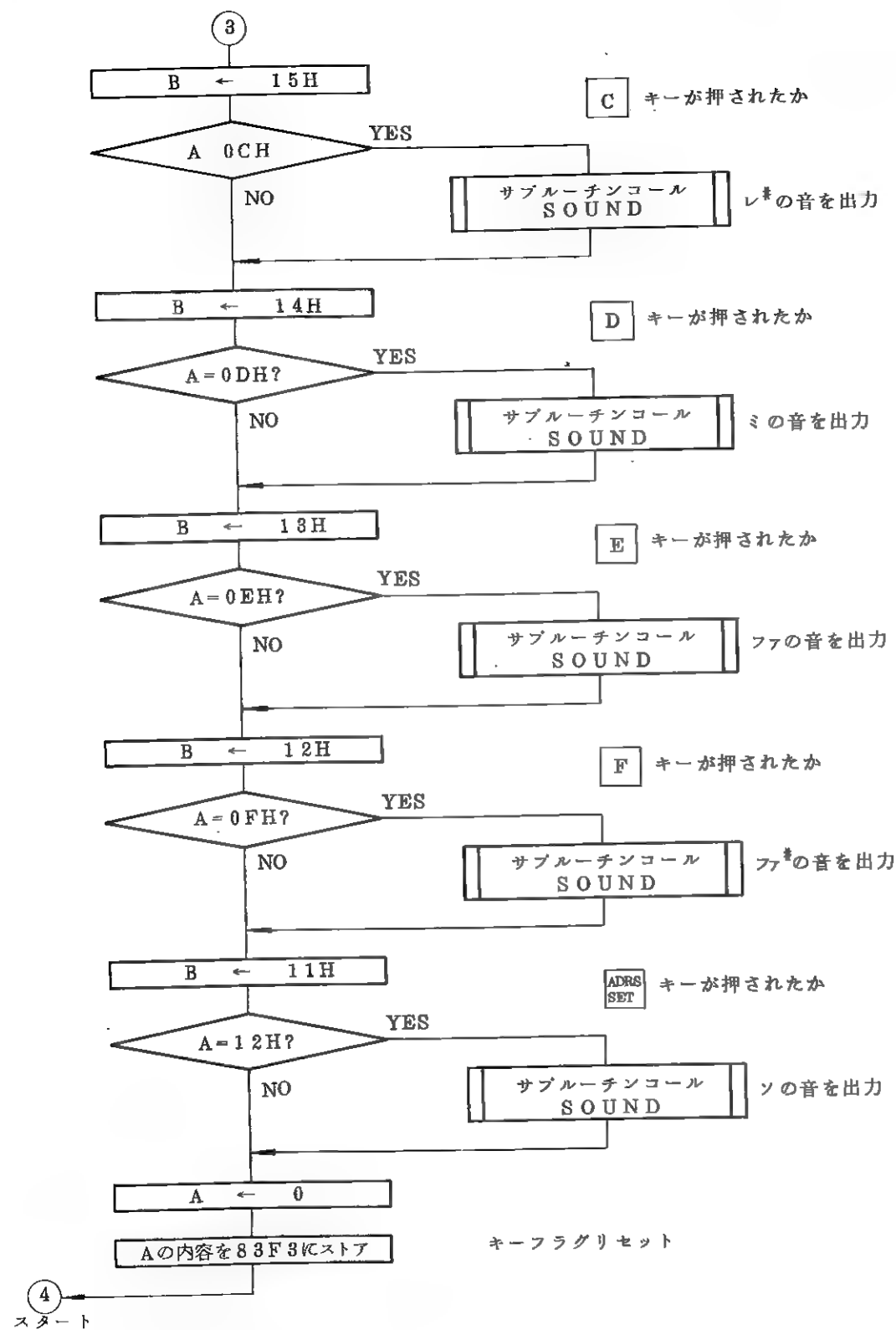
メインルーチンでは、今どのキーが押されているかを検出して、そのキーに対応する音階の周波数パラメータをBレジスタにセットして、サブルーチンをコールしています。

### 4.2 フローチャート

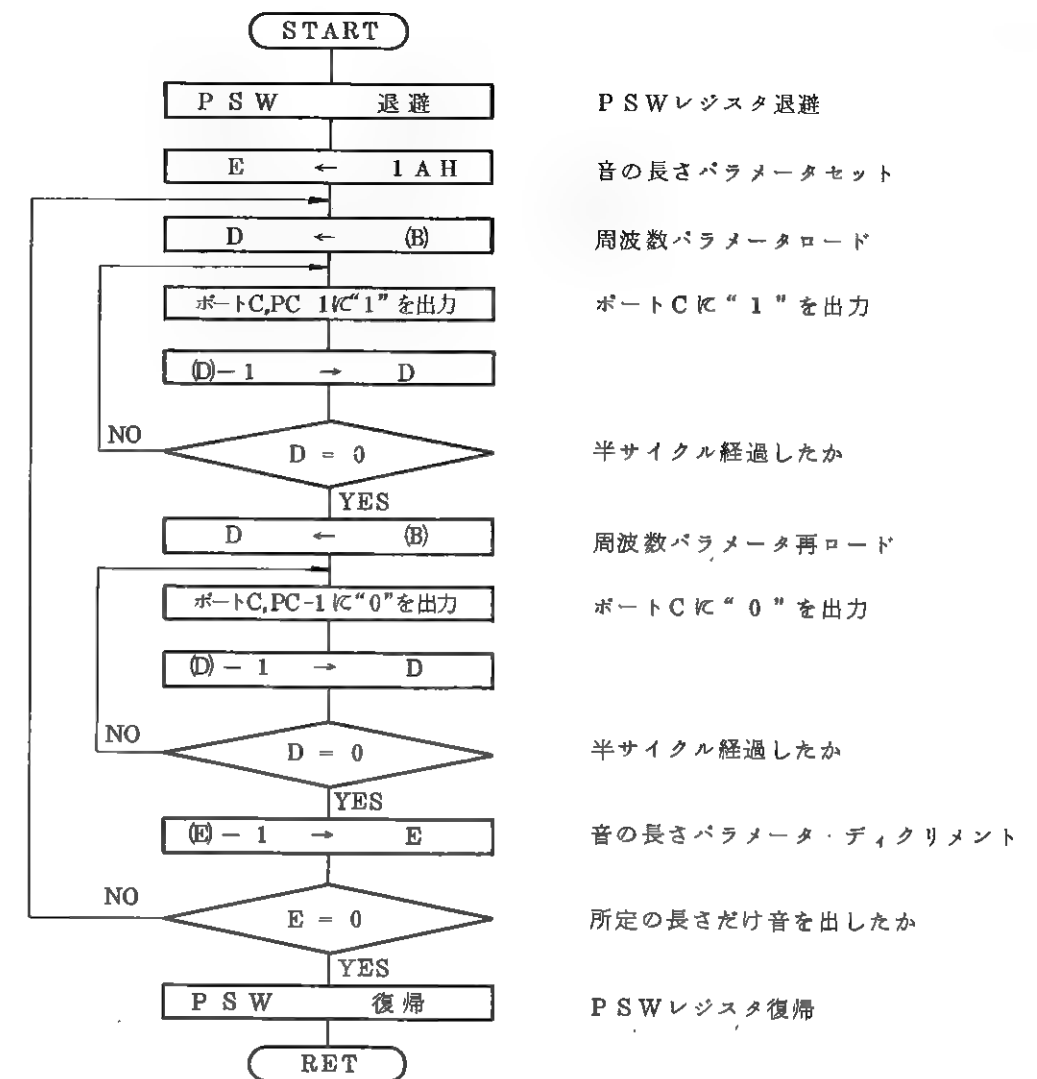
#### (1) メインルーチン







## (2) サブルーチン SOUND



# 4.3 コーディング例

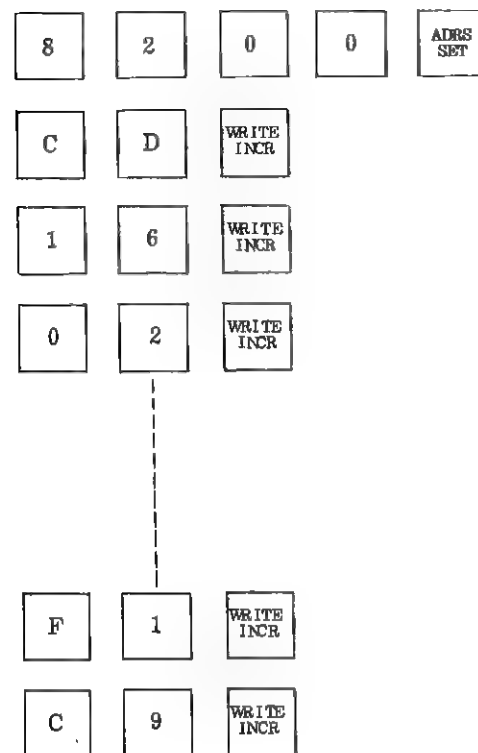
ライン	アドレス	ラベル	ニーモニック	オブジェクトコード
00	8200	START:	CALL KEYIN	CD 16 02
01	3		MVI B, 33H	06 33
02	5		CPI 0	FE 00
03	7		CZ SOUND	CC 96 82
04	A		MVI B, 31H	06 31
05	C		CPI 1	FE 01
06	E		CZ SOUND	CC 96 82
07	11		MVI B, 2DH	06 2D
08	3		CPI 2	FE 02
09	5		CZ SOUND	CC 96 82
10	8		MVI B, 2BH	06 2B
11	A		CPI 3	FE 03
12	C		CZ SOUND	CC 96 82
13	F		MVI B, 28H	06 28
14	21		CPI 15H	FE 15
15	3		CZ SOUND	CC 96 82
16	6		MVI B, 26H	06 26
17	8		CPI 4	FE 04
18	A		CZ SOUND	CC 96 82
19	D		MVI B, 24H	06 24
20	F		CPI 5	FE 05
21	31		CZ SOUND	CC 96 82
22	4		MVI B, 22H	06 22
23	6		CPI 6	FE 06
24	8		CZ SOUND	CC 96 82
25	B		MVI B, 20H	06 20
26	D		CPI 7	FE 07
27	F		CZ SOUND	CC 96 82
28	42		MVI B, 1EH	06 1E
29	4		CPI 13H	FE 13
30	6		CZ SOUND	CC 96 82
31	9		MVI B, 1DH	06 1D
32	B		CPI 8	FE 08
33	D		CZ SOUND	CC 96 82
34	50		MVI B, 1BH	06 1B
35	2		CPI 9	FE 09
36	4		CZ SOUND	CC 96 82
37	7		MVI B, 19H	06 19
38	9		CPI 0AH	FE 0A
39	B		CZ SOUND	CC 96 82
40	E		MVI B, 18H	06 18
41	60		CPI 0BH	FE 0B
42	2		CZ SOUND	CC 96 82
43	5		MVI B, 16H	06 16
44	7		CPI 14H	FE 14

ライン	アドレス	ラベル	ニーモニック	オブジェクトコード
45	8269		CZ SOUND	CC 96 82
46	C		MVI B, 15H	06 15
47	E		CPI 0CH	FE 0C
48	70		CZ SOUND	CC 96 82
49	3		MVI B, 14H	06 14
50	5		CPI 0DH	FE 0D
51	7		CZ SOUND	CC 96 82
52	A		MVI B, 13H	06 13
53	C		CPI 0EH	FE 0E
54	E		CZ SOUND	CC 96 82
55	31		MVI B, 12H	06 12
56	3		CPI 0FH	FE 0F
57	5		CZ SOUND	CC 96 82
58	8		MVI B, 11H	06 11
59	A		CPI 12	FE 12
60	C		CZ SOUND	CC 96 82
61	F		XRA A	AF
62	90		STA 83F3H	32 F3 83
63	3		JMP START	C3 00 82
64	6	SOUND:	PUSH PSW	F5
65	7		MVI E, 1AH	1E 1A
66	9	LOOP1:	MOV D, B	50
67	A	LOOP2:	MVI A, 2	3E 02
68	C		OUT 2	D3 02
69	E		DCR D	15
70	F		JNZ LOOP2	C2 9A 82
71	A2		MOV D, B	50
72	3	LOOP3:	XRA A	AF
73	4		OUT 2	D3 02
74	6		DCR D	15
75	7		JNZ LOOP3	C2 A8 82
76	A		DCR E	1D
77	B		JNZ LOOP1	C2 99 82
78	E		POP PSW	F1
79	F		RET	C9



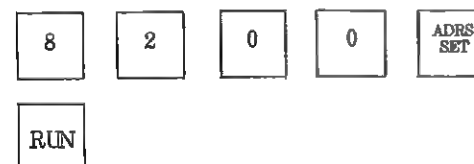
#### 4.4 プログラミングおよび実行方法

4.3 に書かれているオブジェクトコードを所定のメモリに書き込んで行きます。



プログラムの書き込みが終了し、書き込みエラーがないことを確認したならば、次のキーコマンドによりプログラムを実行します。

なお、音を出すためのオーディオアンプの接続法は2.5と全く同じです。



#### 4.5 キーボードと音階との対応

ド	_____	0
ド#	_____	1
レ	_____	2
レ#	_____	3
ミ	_____	WRITE INCR
ファ	_____	4
ファ#	_____	5
ソ	_____	6
ソ#	_____	7
ラ	_____	READ DECR
ラ#	_____	8
シ	_____	9
ド	_____	A
ド#	_____	B
レ	_____	READ INCR
レ#	_____	C
ミ	_____	D
ファ	_____	E
ファ#	_____	F
ソ	_____	ADRS SET

## 第5章 音楽の自動演奏プログラム

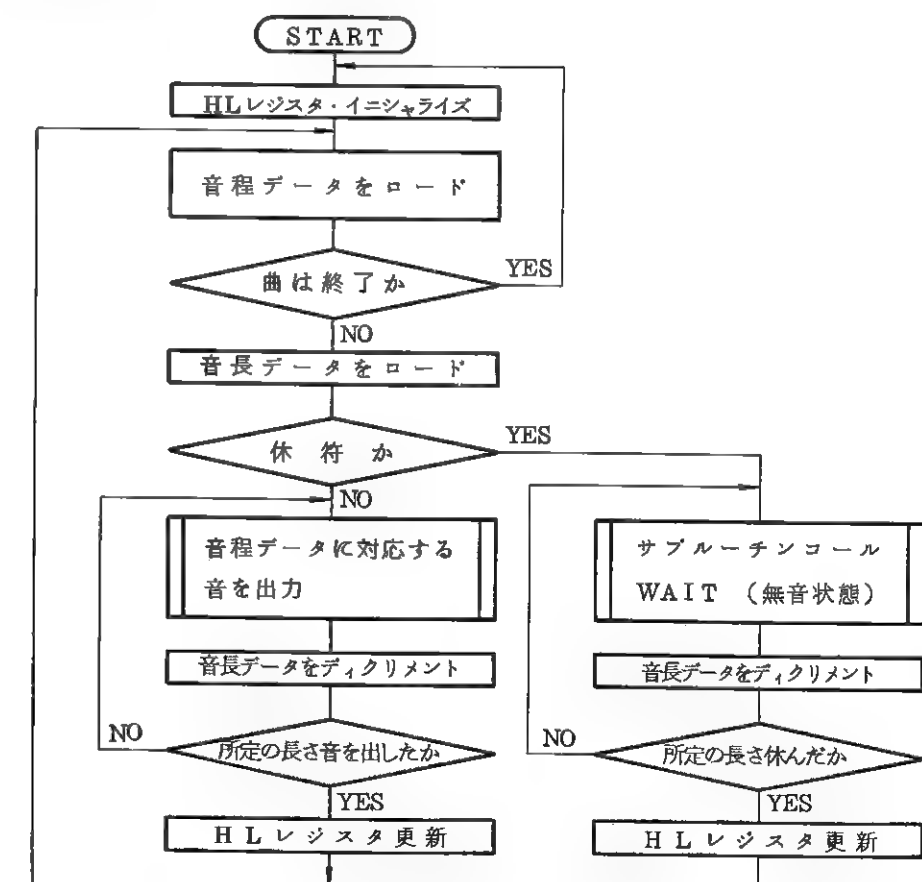
### 5.1 概 要

第4章の電子オルガン・プログラムの応用として音楽を連続して自動演奏するプログラム例を示します。

このプログラムは、あらかじめ演奏させる曲をこのプログラムのフォーマットに従って所定のデータに変換してプログラムのデータエリアに格納しておきます。

プログラムはデータエリアに格納されているデータを参照しながら、曲を演奏して行きます。

### 5.2 概略のフローチャート





## 5.4 コーディング例

ライン	アドレス	レベル	ニーモニック	オブジェクトコード
00	8200	START :	LXI H, 8250H	21 50 82
01	3		MOV A, M	7E
02	4		ANA A	A7
03	5		JZ START	CA 00 82
04	8		INX H	23
05	9		MOV C, M	4E
06	A		MOV B, A	47
07	B		RLC	07
08	C		JC WAIT	DA 3C 82
09	F	LOOP :	PUSH H	E5
10	10		CALL SOUND	CD 1C 82
11	3		POP H	E1
12	4		DCR C	0D
13	5		JNZ LOOP	C2 0F 82
14	8		INX H	23
15	9		JMP START+3	C3 03 82
16	C	SOUND :	LXI H, 35FFH	21 FF 35
17	F	S1 :	MOV D, B	50
18	20		MVI A, 2	3E 02
19	2		OUT 2	D3 02
20	4	S2 :	DCX H	2B
21	5		MOV A, H	7C
22	6		ANA A	A7
23	7		RZ	C8
24	8		DCR D	15
25	9		JNZ S2	C2 24 82
26	C		MOV D, B	50
27	D		MVI A, 0	3E 00
28	F		OUT 2	D3 02
29	31	S3 :	DCX H	2B
30	2		MOV A, H	7C
31	3		ANA A	A7
32	4		RZ	C8
33	5		DCR D	15
34	6		JNZ S3	C2 31 82
35	9		JMP S1	C3 1F 82
36	C	WAIT :	PUSH H	E5
37	D		LXI H, 50FFH	21 FF 50
38	40		DCX H	2B
39	1		MOV A, H	7C
40	2		ANA A	A7
41	3		JNZ \$-3	C2 40 82
42	6		DCR C	0D
43	7		JNZ WAIT+1	C2 3D 82
44	A		POP H	E1
45	B		INX H	23
46	C		JMP START+3	C3 03 82

## 5.5 楽譜データの作成

このプログラムでは、一つの音を、連続した番地に格納されている2ワード(8ビット×2)のデータによって作り出しています。

下位の番地に格納されるデータを音程パラメータ、上位の番地に格納されるデータを音長パラメータと呼ぶことにします。

音程パラメータは、その音が音階のどの音なのかを示し、音長パラメータはその音がどのくらいの長さ鳴りつづけるかを示しています。

### (1) 音程パラメータ

音程パラメータは、発生する音の周波数(パルス発生プログラムの時定数)を設定する時に参照されるパラメータです。

本プログラムでは、次のような値を設定することによって、各音階に対応する音を発生します。

音階	音程パラメータ(16進数)
ド	3 3
ド#	3 1
レ	2 D
レ#	2 B
ミ	2 8
ファ	2 6
ファ#	2 4
ソ	2 2
ソ#	2 0
ラ	1 E
ラ#	1 D
シ	1 B
ド	1 9
ド#	1 8
レ	1 6
レ#	1 5
ミ	1 4
ファ	1 3
ファ#	1 2
ソ	1 1

又、音程パラメータは、その最上位ビットを"1"にすることによって、休符を表わすことができます。

1 × × × × × × × × ..... 休符

## (2) 音長パラメータ

このプログラムは、実際に音を発生させる部分をサブルーチン(SOUND)として持っています。

このサブルーチンは、音程パラメータにより決定された周波数の音を、サブルーチン内でセットされた時定数による時間、発生させます。

一方メインルーチンにおいては、音長パラメータによってセットされた回数だけこのサブルーチンをコールするようになっているために、音長パラメータに値をセットすることにより、サブルーチン内で決定された単位時間の整数倍の時間音を発生させることができるわけです。

従って、サブルーチン内の時定数を変えることにより、楽符を変えることなく曲のテンポを変えることもできます。

この時、定数はプログラムリスト 16ライン及び87ラインにおいてHレジスタにセットされる値により決定され、その値が大きくなれば、曲のテンポはおそくなります。(このプログラム例では35(16進)となっています。)

## (3) 楽譜作成例

音程パラメータ、音長パラメータは8250番地より順に書き込んで行きます。又、曲の終わりは、一定の休符を入れた後、1ワード"00"を書いておくことによりその曲をくり返し演奏させることができます。

## 例 ドレミの歌

アドレス	楽符	アドレス	楽符	アドレス	楽符
8250	33	827E	22	82AC	19
	03		01		07
52	2D	80	26	AE	1B
	01		01		01
54	28	82	1E	B0	1E
	03		08		02
56	33	84	22	B2	26
	01		03		02
58	28	86	33	B4	1B
	02		01		02
5A	33	88	2D	B6	22
	02		01		02
5C	28	8A	28	B8	19
	04		01		06
5E	2D	8C	26	BA	FF
	03		01		08
60	28	8E	22	BC	00
	01		01		— エンドコード
62	26	90	1E		
	02		08		
64	28	92	1E		
	01		08		
66	2D	94	2D		
	01		01		
68	26	96	28		
	08		01		
6A	28	98	26		
	03		01		
6C	26	9A	22		
	01		01		
6E	22	9C	1E		
	03		01		
70	28	9E	1B		
	01		08		
72	22	A0	1B		
	02		03		
74	28	A2	28		
	02		01		
76	22	A4	26		
	04		01		
78	26	A6	22		
	03		01		
7A	22	A8	1E		
	01		01		
7C	1E	AA	1B		
	02		01		

## 第6章 無限音階プログラム

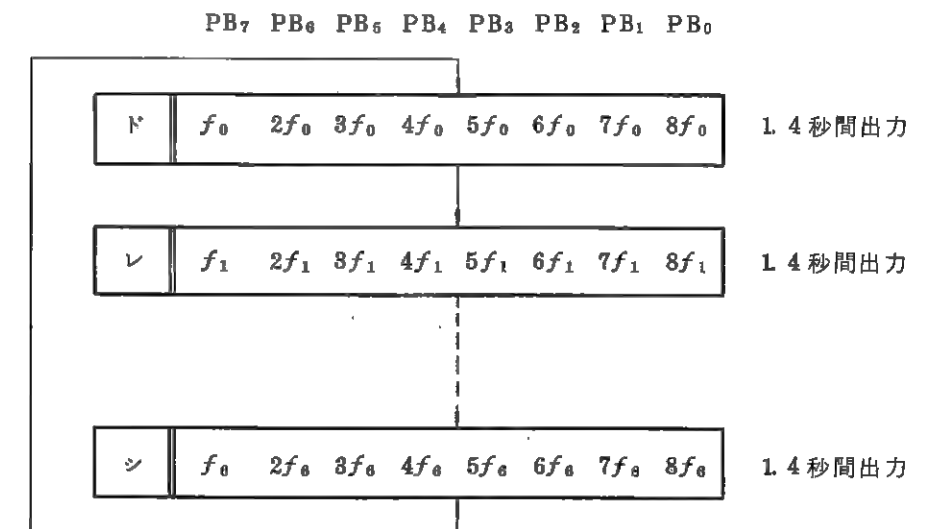
### 6.1 概 要

PPI( $\mu$ PD8255)のポートBに接続したオーディオ・アンプに無限音階を出力するプログラム例を示します。

このプログラムはそれぞれ1オクターブずつ離れたオーディオ周波数のパルスをPPIのPB<sub>0</sub>～PB<sub>7</sub>に8オクターブ同時に出力し(例えば“ド”の音の場合はPB<sub>0</sub>にある“ド”の音, PB<sub>1</sub>はPB<sub>0</sub>より1オクターブ下の“ド”の音, PB<sub>2</sub>にはさらに1オクターブ下の“ド”の音……), これらをそれぞれ1音階ずつ繰り返します。

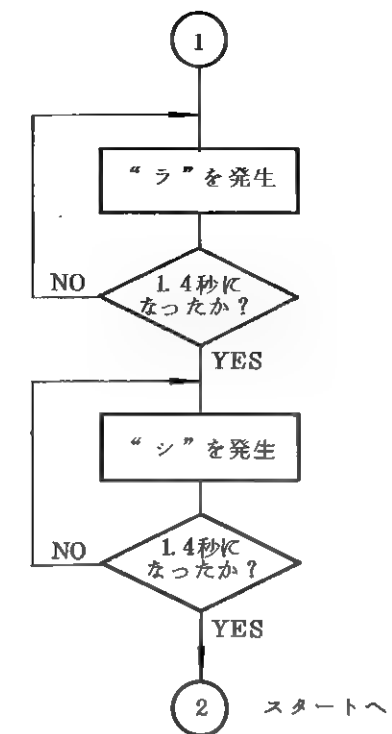
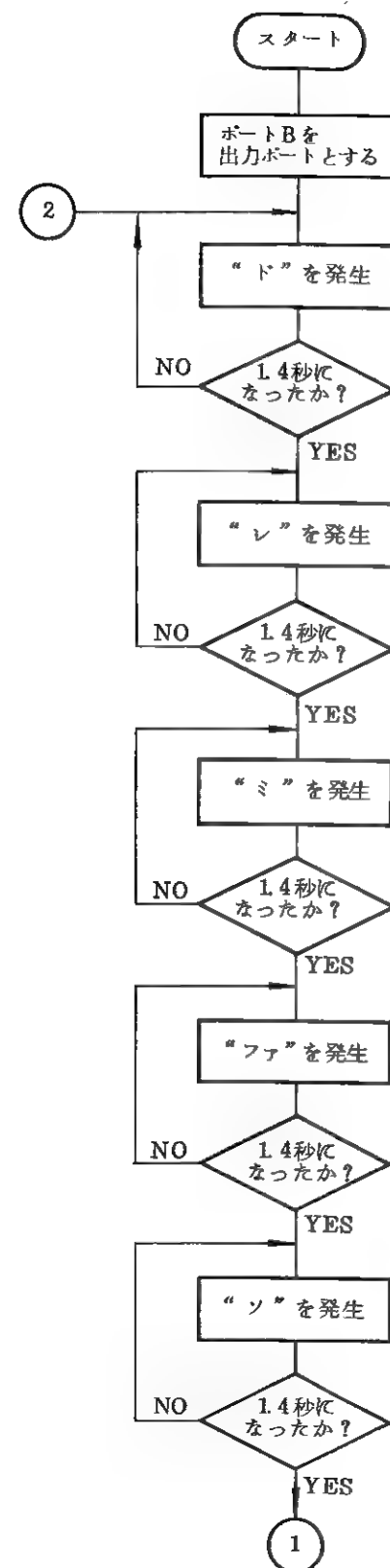
このパルスは合成されてオーディオ・アンプに入力し, 音声として出力されます。

合成出力は各音ごとに8オクターブの音声を含むため, 人間の耳にはその人が注目した音から無限に上昇する音階となって聞こえます。

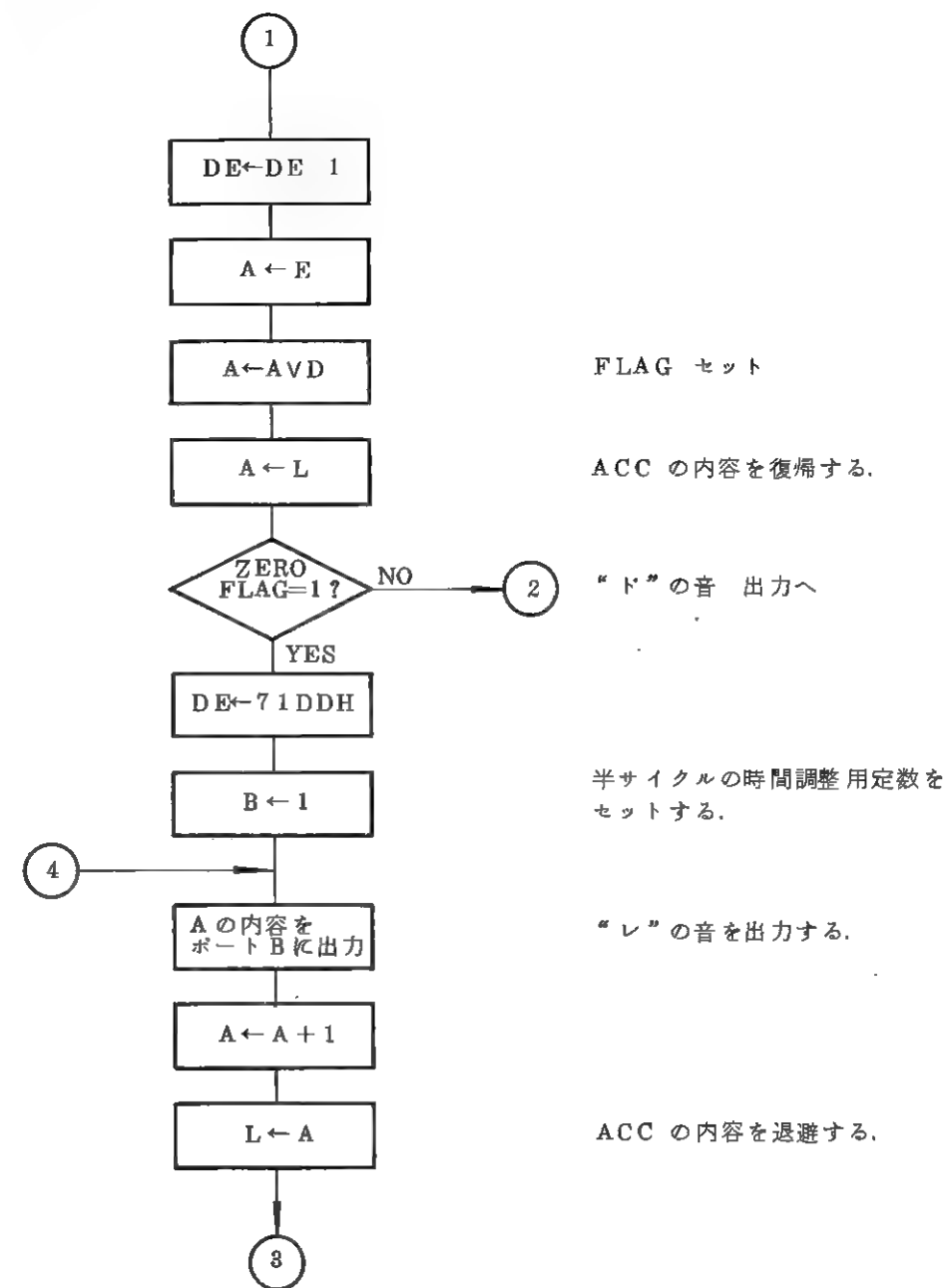
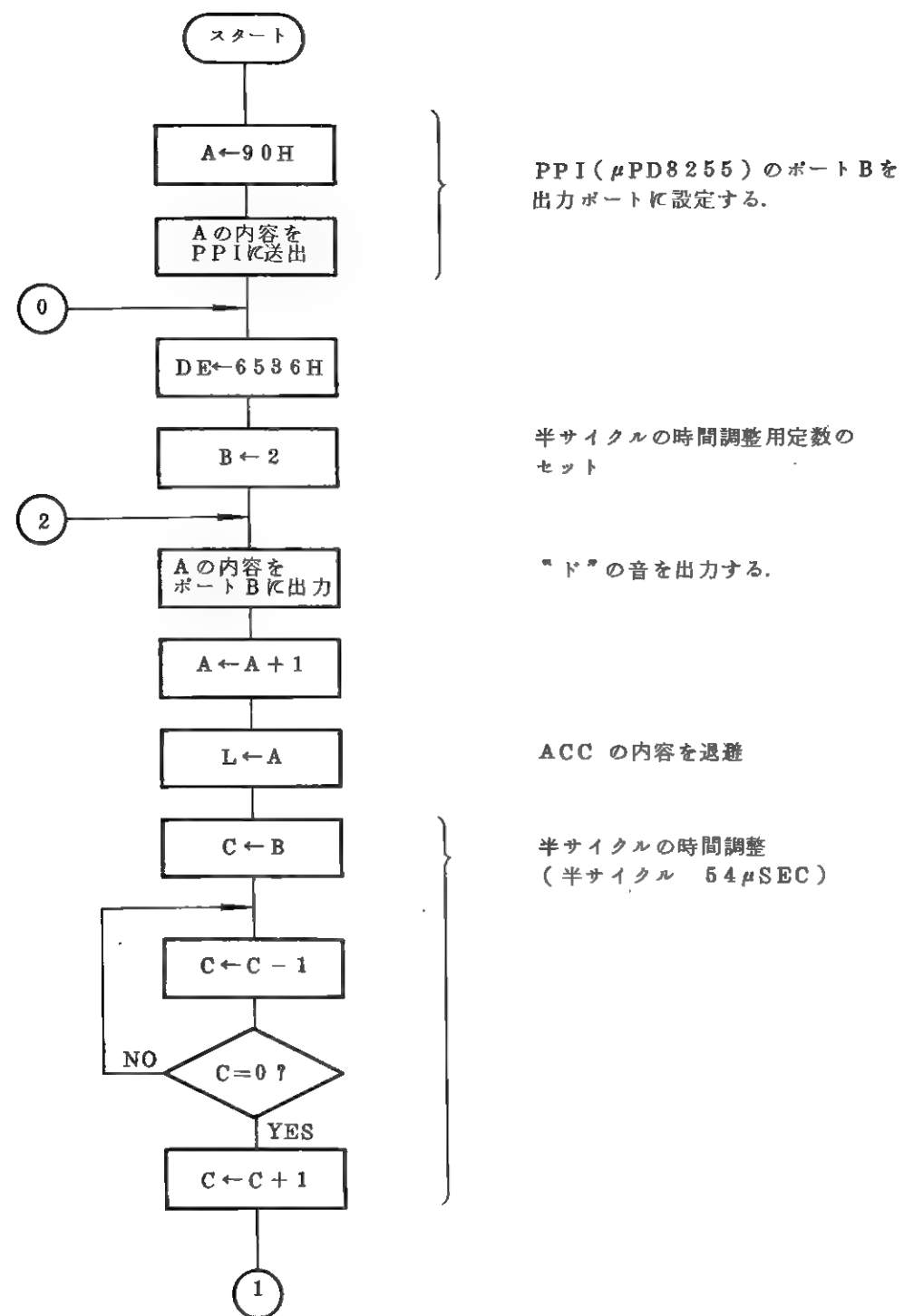


(注)  $f_0$  はある音階の“ド”の音を示す,  $f_1 \dots f_8$  も同様。

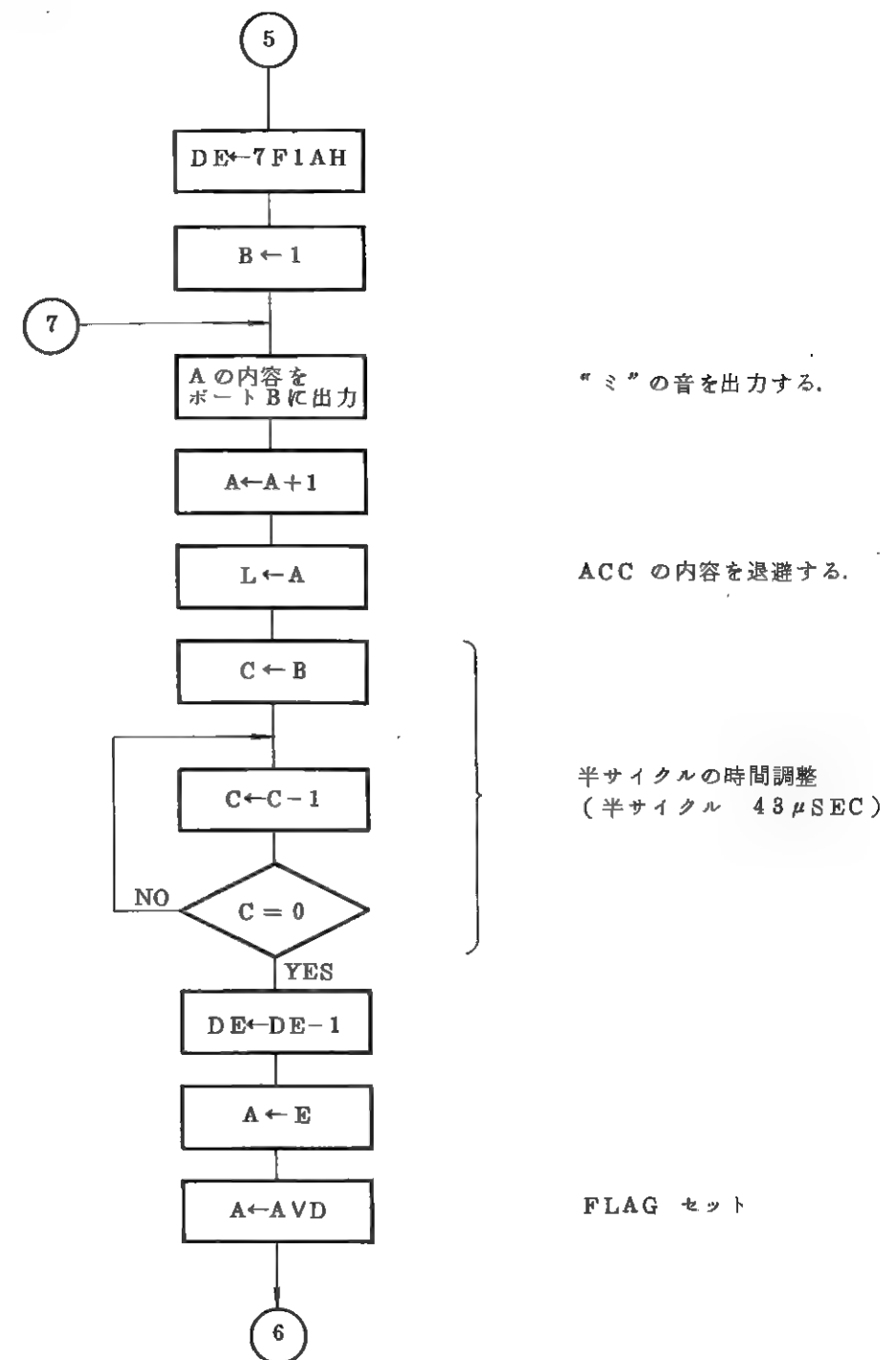
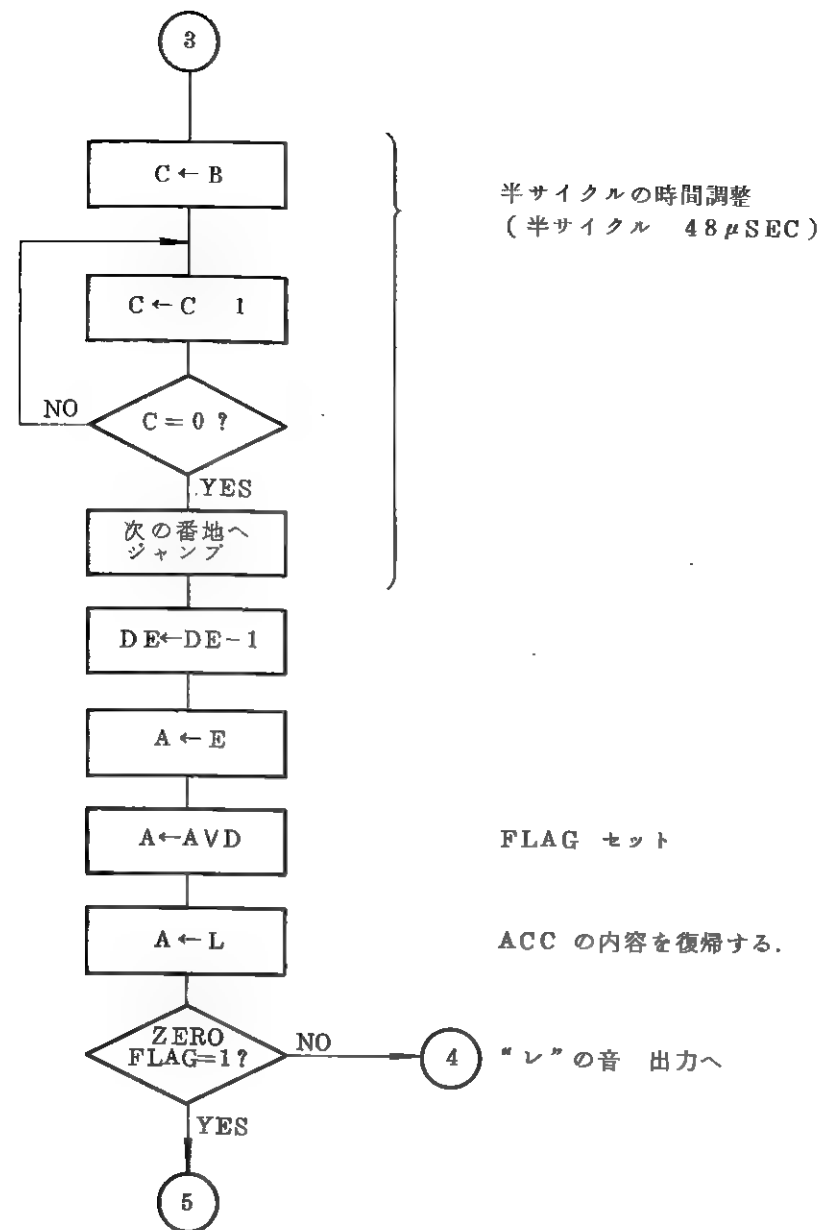
## 6.2 概略のフローチャート

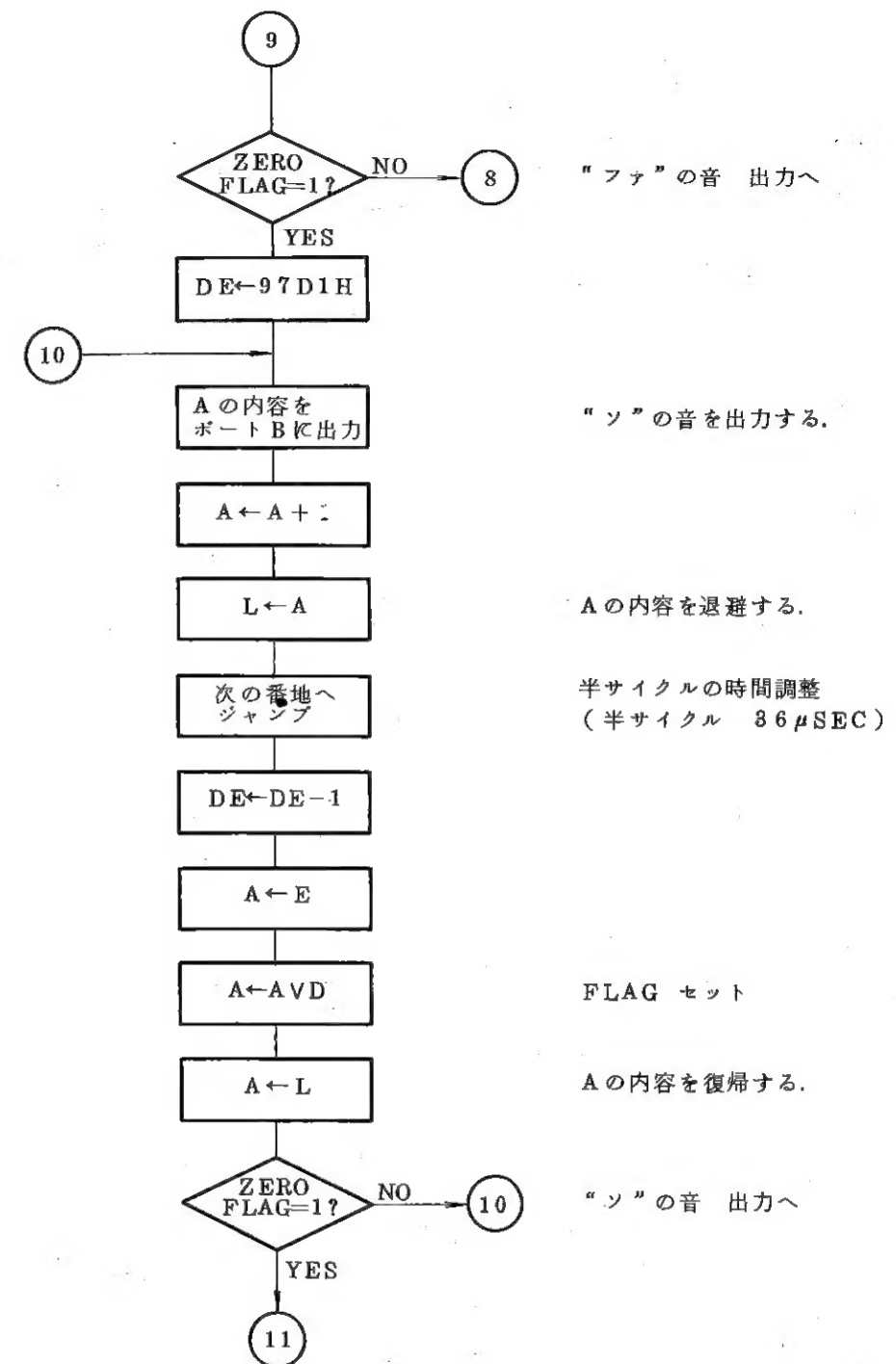
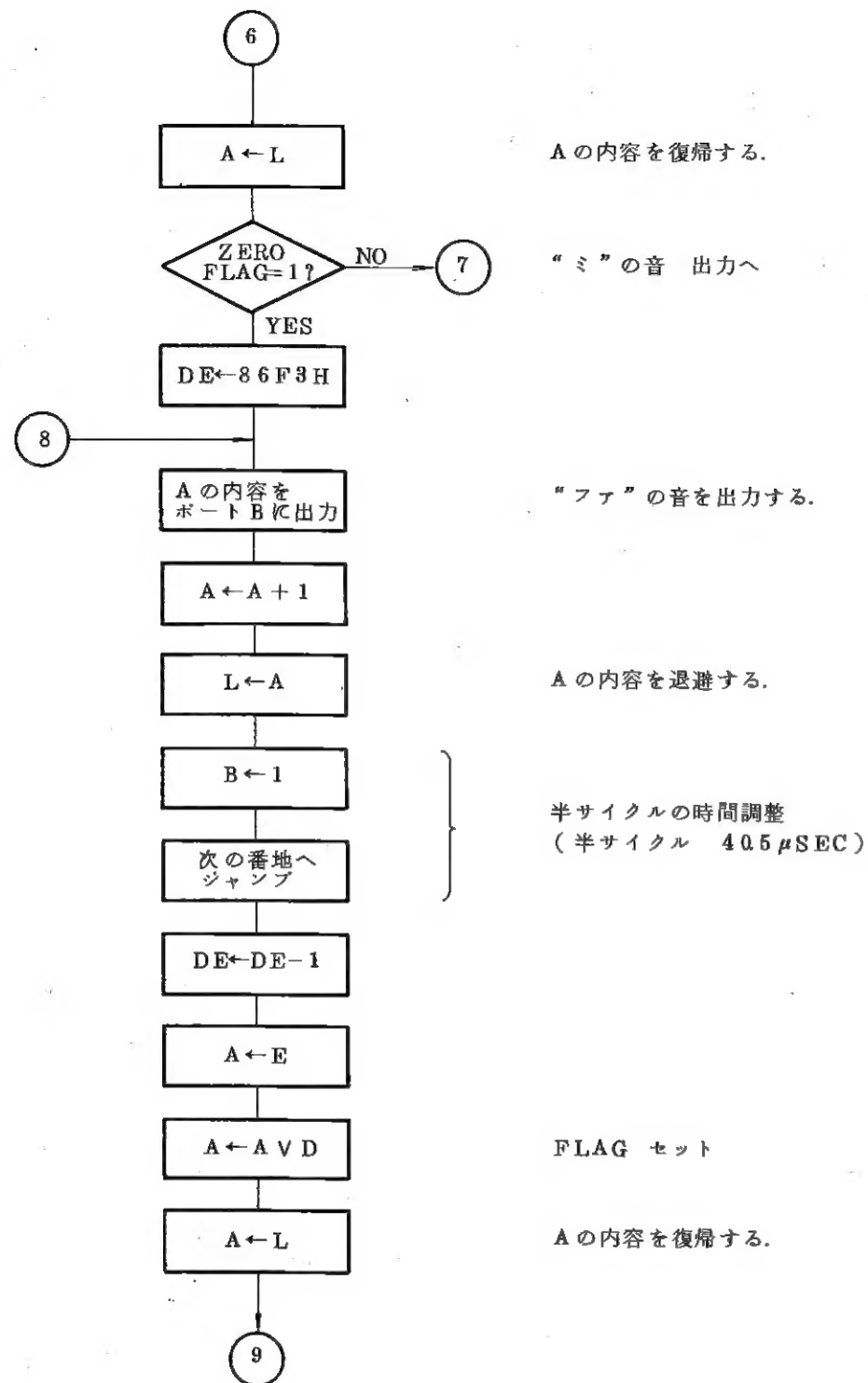


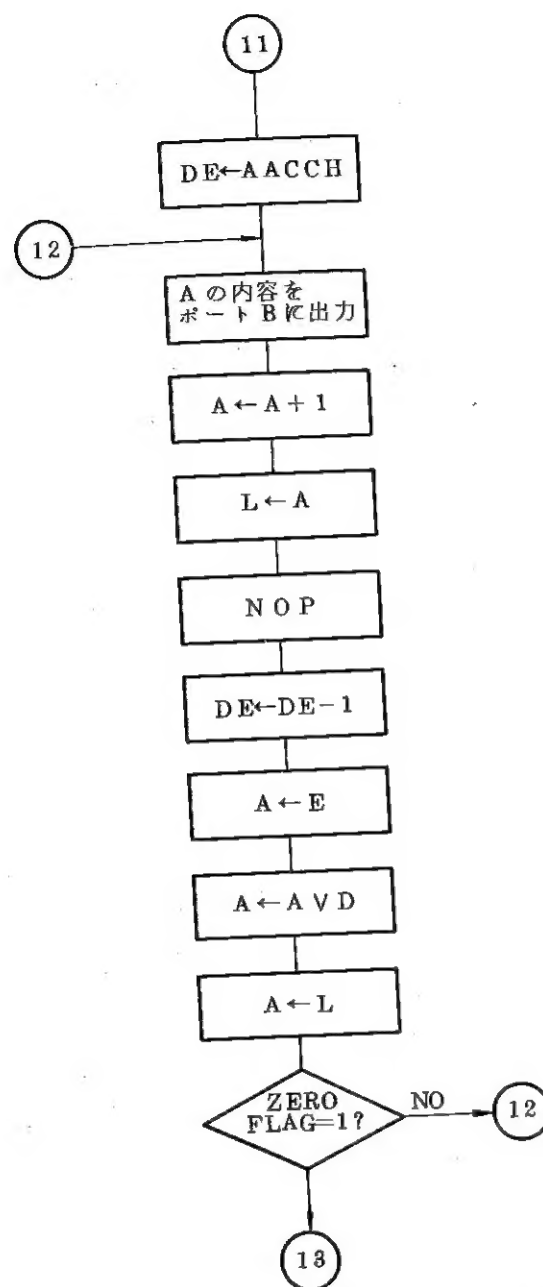
### 6.3 詳細なフローチャート









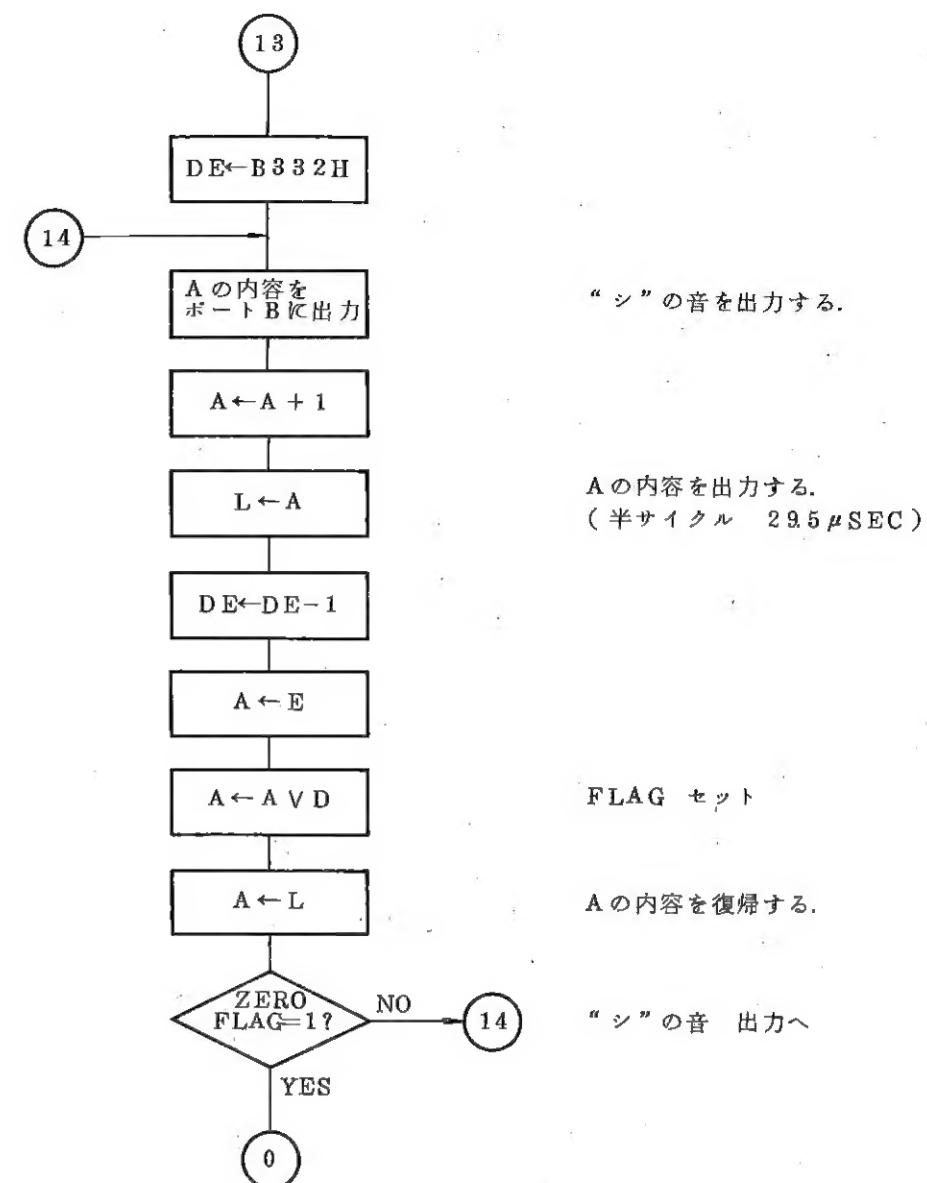


“ラ”の音を出力する。

半サイクルの時間調整  
(半サイクル 32 μSEC)

FLAG セット

Aの内容を復帰する。



“シ”の音を出力する。

Aの内容を出力する。  
(半サイクル 29.5 μSEC)

FLAG セット

Aの内容を復帰する。

“シ”の音 出力へ

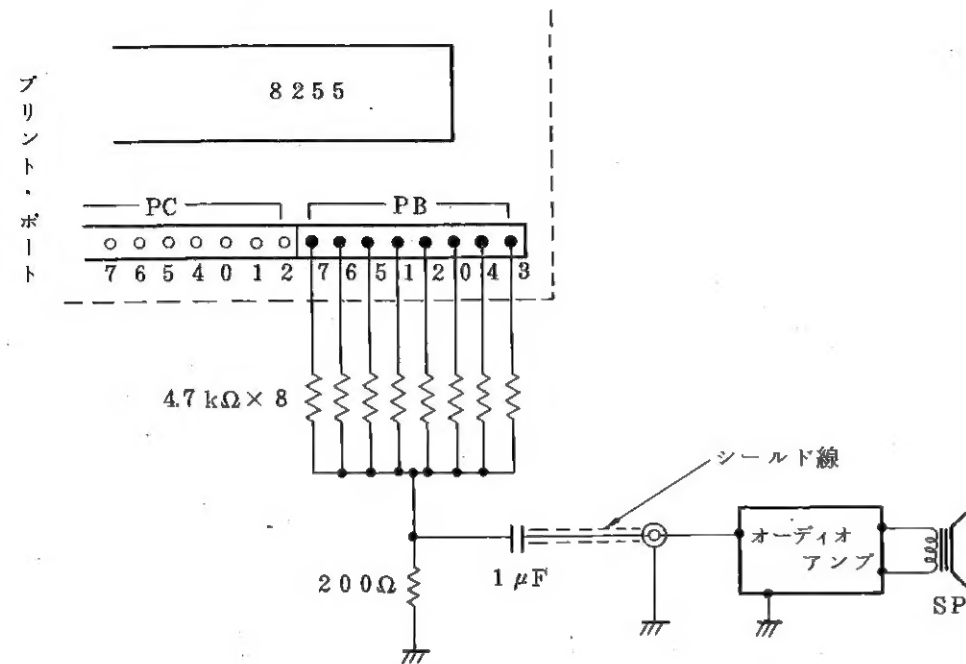
# 6.4 コーディング例

ライン	アドレス	ラベル	ニーモニック	オブジェクトコード
00	8200	ST :	MVI A, 90H	3E 90
01	02		OUT 03H	D3 03
02	04		LXI D, 6536H	11 36 65
03	07		MVI B, 2	06 02
04	09	DO :	OUT 01H	D3 01
05	0B		INR A	3C
06	0C		MOV L, A	6F
07	0D		MOV C, B	48
08	0E		DCR C	0D
09	0F		JNZ \$-1	C2 0E 82
10	12		INR C	0C
11	13		DCX D	1B
12	14		MOV A, E	7B
13	15		ORA D	B2
14	16		MOV A, L	7D
15	17		JNZ DO	C2 09 82
16	1A		LXI D, 71DDH	11 DD 71
17	1D		MVI B, 1	06 01
18	1F	RE :	OUT 01H	D3 01
19	21		INR A	3C
20	22		MOV L, A	6F
21	23		MOV C, B	48
22	24		DCR C	0D
23	25		JNZ \$-1	C2 24 82
24	28		JMP \$+3	C3 2B 82
25	2B		DCX D	1B
26	2C		MOV A, E	7B
27	2D		ORA D	B2
28	2E		MOV A, L	7D
29	2F		JNZ RE	C2 1F 82
30	32		LXI D, 7F1AH	11 1A 7F
31	35		MVI B, 1	06 01
32	37	MI :	OUT 01H	D3 01
33	39		ADI 1	C6 01
34	3B		MOV L, A	6F
35	3C		MOV C, B	48
36	3D		DCR C	0D
37	3E		JNZ \$-1	C2 3D 82
38	41		DCX D	1B
39	42		MOV A, E	7B
40	43		ORA D	B2
41	44		MOV A, L	7D

ライン	アドレス	ラベル	ニーモニック	オブジェクトコード
42	8245		JNZ MI	C2 37 82
43	48		LXI D, 86F3H	11 F3 86
44	4B	FA :	OUT 01H	D3 01
45	4D		INR A	3C
46	4E		MOV L, A	6F
47	4F		MVI B, 1	06 01
48	51		JMP \$+3	C3 54 82
49	54		DCX D	1B
50	55		MOV A, E	7B
51	56		ORA D	B2
52	57		MOV A, L	7D
53	58		JNZ FA	C2 4B 82
54	5B		LXI D, 97D1H	11 D1 97
55	5E	SO :	OUT 01H	D3 01
56	60		INR A	3C
57	61		MOV L, A	6F
58	62		JMP \$+3	C3 65 82
59	65		DCX D	1B
60	66		MOV A, E	7B
61	67		ORA D	B2
62	68		MOV A, L	7D
63	69		JNZ SO	C2 5E 82
64	6C		LXI D, 0AACCH	11 CC AA
65	6F	RA :	OUT 01H	D3 01
66	71		INR A	3C
67	72		MOV L, A	6F
68	73		NOP	00
69	74		DCX D	1B
70	75		MOV A, E	7B
71	76		ORA D	B2
72	77		MOV A, L	7D
73	78		JNZ RA	C2 6F 82
74	7B		LXI D, 0B332H	11 32 B3
75	7E	SHI :	OUT 01H	D3 01
76	80		INR A	3C
77	81		MOV L, A	6F
78	82		DCX D	1B
79	83		MOV A, E	7B
80	84		ORA D	B2
81	85		MOV A, L	7D
82	86		JNZ SHI	C2 7E 82
83	89		JMP ST+4	C3 04 82

## 6.5 オーディオ・アンプの接続方法

PPI のポート B に抵抗を接続してそれぞれの周波数を合成し、カップリング・コンデンサを介してオーディオ・アンプに接続します。



## 6.6 プログラミングおよび実行方法

コーディング・リスト上のオブジェクト・コードを所定のメモリに書き込みます。

8	2	0	0	ADRS SET
3	E	WRITE INCR		
9	0	WRITE INCR		
...				
0	4	WRITE INCR		
8	2	WRITE INCR		

プログラムの書込みが終了し、書き込みエラーがないことを確認したら次のコマンドで実行します。

8	2	0	0	ADRS SET
RUN				